AD NUMBER
AD253273
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; JAN 1961. Other requests shall be referred to Air Force Cambridge Research Laboratories, Bedford Massachusetts.**
AUTHORITY
AFCRL ltr, 3 Nov 1971

AFCRL-108

21600

393 350

SYNTACTIC ANALYSIS IN AUTOMATIC TRANSLATION

MURRAY E. SHERRY

JANUARY 1961

ELECTRONICS RESEARCH DIRECTORATE
AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
AIR FORCE RESEARCH DIVISION
AIR RESEARCH AND DEVELOPMENT COMMAND
UNITED STATES AIR FORCE
BEDFORD MASSACHUSETTS

AFCRL-108

[redacted]

SYNTACTIC ANALYSIS IN AUTOMATIC TRANSLATION

Murray E. Sherry

This report was originally published as Report NSF-5 on Mathematical Linguistics and Automatic Translation to the National Science Foundation by the Computation Laboratory of Harvard University.

Project 5632

Task 56325

January 1961

Computer and Mathematical Sciences Laboratory
Electronics Research Directorate
Air Force Cambridge Research Laboratories
Air Force Research Division (ARDC)
United States Air Force
Bedford, Massachusetts

SYNTACTIC ANALYSIS IN AUTOMATIC TRANSLATION

A thesis presented

by

Murray Elliot Sherry

to

The Division of Engineering and Applied Physics

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Applied Mathematics

Harvard University

Cambridge, Massachusetts

August 1960

PREFACE

A new morphological word-by-word analysis technique and a new syntactic sentence-by-sentence analysis method applicable to the automatic translation of Russian to English are presented in this thesis.

The writer is deeply indebted to Professor Anthony Oettinger for his inspiration and guidance, which greatly influenced the course of the work. He is also indebted to Professor Peter Calingaert and Dr. Vincent Giuliano for reading the manuscript and for valuable suggestions and criticism. The assistance and advice of other members of the staff of the Computation Laboratory, especially Stefanie von Susich, William Foust, David Isenberg, and Barbara Maggs, are gratefully acknowledged. The writer is also indebted to Professor Howard Aiken, who originally proposed the study of automatic translation at Harvard, for his continued support of this activity. The continued interest of Professors Joshua Whatmough and Roman Jakobson in the work of automatic translation is also appreciated.

The present research, undertaken while the writer was on assignment at Harvard University from the United States Air Force, was supported in part by the National Science Foundation and the United States Air Force.

The editorial work on this thesis was coordinated by Helen Sharrow and Joyce McDaniel. Eileen Seaward, Erma Bushek, and Joan Kelley typed the manuscript and many of the plates, and William Minty drew the figures. The photographic work was done by Paul Donaldson. Robert Burns, Louis Kloff, and Joseph Lewko assisted in the preparation of the negatives. To all these people the writer wishes to express his appreciation for their assistance.

NOTE ON THE REVISED REPRINT

The writer wishes to take this opportunity to express his gratitude to the National Bureau of Standards, and in particular to Mrs. Ida Rhodes, for making available her original work on a predictive syntactic analysis technique. This work provided a basis for a large segment of this report. It is necessary for the writer to apologize for the omission of this acknowledgement from the original publication of this report.

The writer is grateful to his associates who have taken the trouble to point out typographical errors in the original work which have been corrected in this printing.

Murray E. Sherry
Air Force Cambridge Research
Laboratories
1 January 1961

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF TABLES

LIST OF TABLES (continued)

SYNOPSIS

This thesis is concerned with a method for the syntactic analysis of Russian sentences. Applied to automatic translation, this method is divided into a morphological word-by-word phase and a syntactical sentence-by-sentence phase.

An idealized canonical stem dictionary is presented, and its significant lexicographic properties are pointed out. This idealized dictionary then serves as a basis for evaluating the actual Harvard Automatic Dictionary. Aspects of morphological analysis of the Russian language and the series of programs written to carry it out are described. To explain the practical problems encountered in an experimental syntactic analysis program, of which a detailed description is given, a new model of natural language is introduced. A more detailed outline of this thesis is given in Chapter 1.

The idealized canonical stem dictionary, the method of morphological analysis of Russian, the construction of the new model of natural language and substantial aspects of the realization of an operating experimental syntactic analysis program represent efforts of the writer.

SYNTACTIC ANALYSIS IN AUTOMATIC TRANSLATION

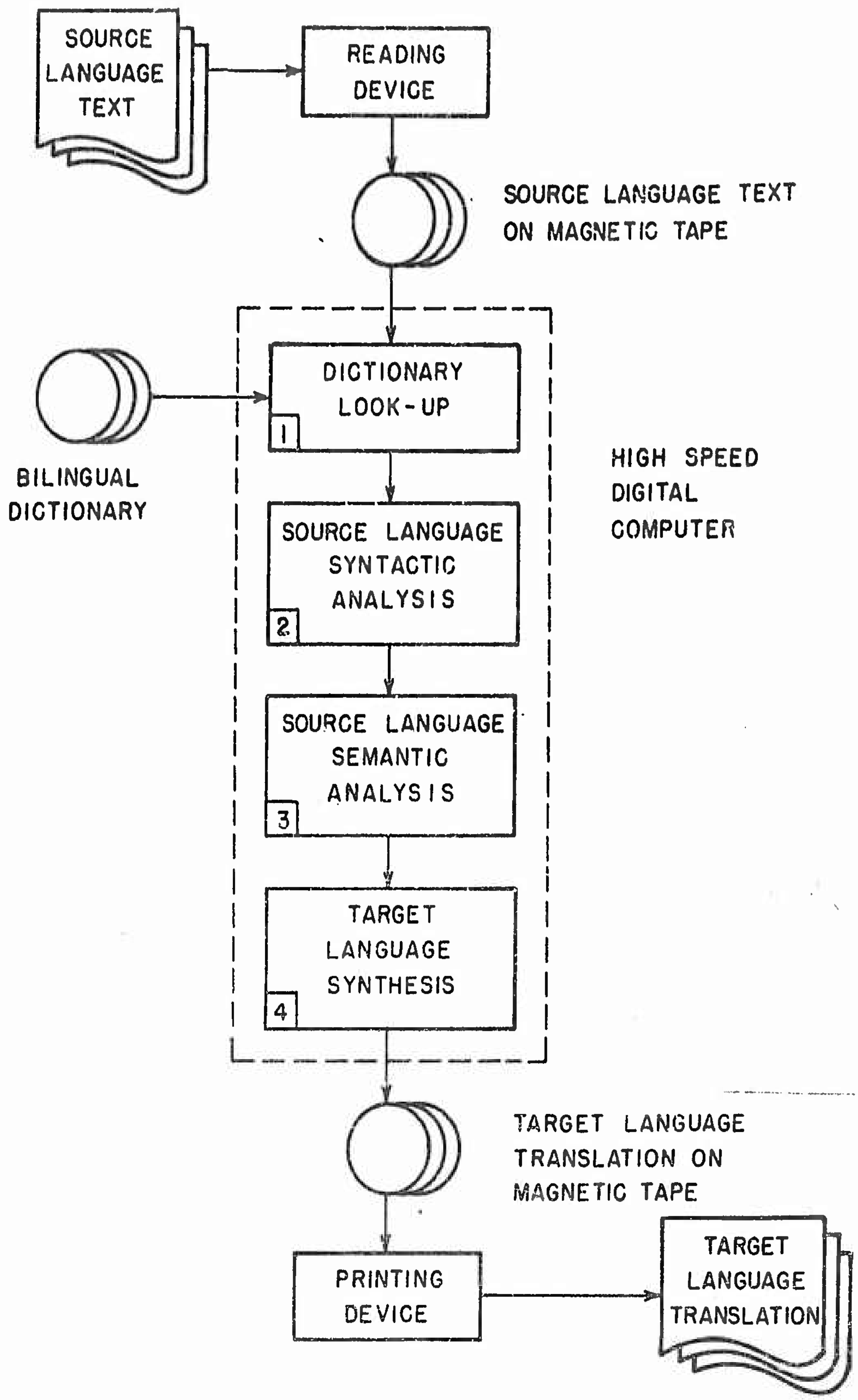
Chapter 1

INTRODUCTION

A block diagram of an idealized system for translating automatically between two languages is given in Fig. 1-1. The text in the source language is first transcribed onto some medium suitable for input to a high speed digital computer. Next, the text is translated into the target language by an appropriate sequence of programs. Finally, the translated text is recorded onto a medium suitable for reading or reproduction.

To prepare a text for processing by a digital computer, it is necessary to transcribe the text onto a magnetic tape or some equivalent medium. Ideally, transcription should be performed by a print-reading machine capable of identifying the various types of letters found on a printed page. At present, the texts which are used for experimental purposes are laboriously typed either onto punched cards or directly onto paper or magnetic tape. At the other end of the process, the output of the computer program can be reproduced singly or in multiple quantities by a number of satisfactory processes. If the recognition of diagrams and pictures is desired, further complications arise at the transcription and the recording steps.

The process of translating a text, as carried out on a digital computer, can be subdivided into four phases: dictionary lookup, syntactic analysis, semantic analysis, and target language synthesis. To look up the words of the source language, a bilingual dictionary, a sequence of programs to control the operation of the dictionary, and a set of programs for correcting and updating the dictionary are necessary. The grammatical roles of the



Idealized Scheme for Automatic Translation

Fig. 1-1

source language words, which are functions of both the lexical characteristics of the individual words and the relationships among the words in a sentence, are determined by the syntactic analysis. In general, more than one target language correspondent is stored in the dictionary entry of a source language word, since the source language word can take on different meanings when used in different contexts. The appropriate meaning of each source language word in its given context is selected by the semantic analysis of the syntactically analyzed text. Finally, the target language correspondents are inflected, rearranged, and appropriate words such as prepositions and articles are added where required.

As an example of the complete process, consider the Russian to English translation of the sentence: Распад каждого атома происходит мгновенно, подобно взрыву. Possible English correspondents of the Russian words in a Russian-English dictionary are given in Table 1-1. The analysis of the sentence that will now be described is idealized, although some sections of the analysis are already in operation and will be discussed in this thesis. An analysis of the individual words, based on their lexical characteristics, is given in Table 1-2.

A syntactic study of the sentence would result in the following analysis. Распад is nominative since it is the subject of the sentence. Каждого is used adjectivally in the genitive possessive noun phrase каждого атома. Происходит is the predicate head and мгновенно is an adverb modifying the verb. Подобно взрыву is an adverbial phrase that modifies the verb.

распад	disintegration, decomposition
каждого	of each (one), of every (one)
атома	atom
происходит	happen, occur, take place, descend
мгновенно	instantaneous, momentary
подобно	like, similar
взрыву	explosion, outburst, burst

English Equivalents for Some Russian
Words in a Russian-English Dictionary

TABLE 1-1

распад	Either nominative or accusative singular, masculine noun.
каждого	Pronoun used adjectively or nominally, either genitive singular and either masculine or neuter, or accusative singular masculine.
атома	Genitive singular masculine noun.
происходит	Third person singular, present tense, indicative verb.
мгновенно	Adverb that can be used as a predicate in place of a verb.
подобно	Adverb that can be used as a predicate in place of a verb.
взрыву	Dative singular masculine noun.

Word-by-Word Analysis of the Sample Sentence

TABLE 1-2

The next phase would select the appropriate English correspondents. "Disintegration" would be selected for распад, "instantaneous" for мгновенно, "like" for подобно, and "explosion" for взрыву. Either alternative could be used for каждого and any of the first three alternatives for происходит.

Three of the English equivalents would be inflected. Since каждого is used adjectivally, the correspondent would be "of each" rather than "of each one". An "s" would be added to an English verb such as "happens" for "happen". Finally, a "ly" would be added to "instantaneous" to indicate the adverbial usage. The English translation would then be:

"Disintegration of each atom happens instantaneously, like explosion."

The translation would be complete if the English articles were included:

"The disintegration of each atom happens instantaneously, like an explosion."

The thesis is chiefly concerned with the second of the four processes of Fig. 1-1. The ability to carry out the experimental analysis is predicated on the existence of an automatic Russian-English dictionary and its associated controlling routines. In this discussion <u>syntactic analysis</u> will include both the morphological word-by-word analysis and the sentence-by-sentence analysis described in the previous example.

The method for producing the morphological analysis can vary over a wide range and is mainly a function of the type of dictionary used. In a full paradigm dictionary†, which has a unique entry for every inflected

† The set of all inflected forms for a given word is called the <u>paradigm</u> of the word.

word form, it is possible to store all the known syntactic information pertinent to each word form directly in the dictionary and read it out whenever the given form is looked up in the dictionary. An alternative is to store a segment of a word form common to all the paradigmatic forms of the word rather than the whole word form. A single dictionary entry can then represent the entire paradigm. Such a dictionary requires much less storage space than a full paradigm dictionary because, as Giuliano has indicated,[1] there is an average of about ten word forms within a Russian paradigm. So long as large-scale storage devices remain extremely expensive to acquire and operate, the latter alternative will seem more attractive.

Since the Harvard Automatic Dictionary, which is a compromise between the two extremes, is a result of the cumulative efforts of a number of investigators over several years, it has become difficult to isolate the essential features of the system from the pieces that have been incorporated to make up for previously encountered shortcomings. An idealized canonical stem dictionary is presented in Chapter 2 to point out, on the one hand, the significant lexicographic details of such a dictionary and to provide, on the other hand, a basis for comparing actual dictionaries and particularly, for evaluating the actual Harvard Automatic Dictionary. The idealized dictionary is described in a mathematical notation in an attempt to ascribe clearly defined characteristics to it. Included in this chapter is a method for the construction of the dictionary and of the individual entries, as well as a method for the morphological analysis of text words.

The author is indebted to D. W. Davies of the National Physical Laboratory, England, for comments which provided a point of departure for the investigation reported in Chapter 2. Mr. Davies visited Cambridge,

Massachusetts, in December, 1959, after he and his staff had studied the previous publications of the Harvard project. To store grammatical information in dictionary entries, Mr. Davies outlined a scheme which is approximately the same as the "entry function vector".

Whereas in a full paradigm dictionary the individual dictionary entries can be precoded with all the grammatical information relevant to each inflected form, this approach is impossible when a stem dictionary is used. Some of the grammatical information in a Russian-English dictionary, such as case and number, is dependent on the word endings. It is therefore necessary to analyze the endings and stem dictionary entries after the look-up process. As many ambiguities as possible are resolved on a word-by-word basis to reduce the burden placed on the more complex sentence-by-sentence syntactic analysis which follows the morphological analysis. The problems involved in the word-by-word analysis are discussed in Chapter 3 and the analysis programs are presented there. In addition, several other programs that have been written to patch the existing dictionary are included in this chapter. The output of these programs is identical with the output of the idealized dictionary described in Chapter 2, although the processes differ greatly in detail.

The method of predictive syntactic analysis is based on the empirical technique for the syntactic analysis of Russian devised by I. Rhodes of the U.S. National Bureau of Standards. The author had the privilege of being introduced to this technique while working with Mrs. Rhodes during the summer of 1959. The technique is based on the premise that many Russian sentences can be analyzed on a left-to-right pass, scanning each word of

the sentence once and in order. The syntactic role of a word in a sentence can be determined from the syntactic roles of the words preceding it. Moreover, on the basis of the analyzed word, it is possible to make further predictions about the syntactic roles of the words which can follow. The predictions are stored in a prediction pool, an approximation to a simple pushdown store, that is, a linear array of storage devices in which information is entered and removed from one end only according to a "last-in-first-out" technique.

In an effort to explain the practical problems arising in the predictive analysis of natural languages, a model of natural language has been developed in Chapter 4. The algorithms which operate on the model language show the essential usefulness of the fundamental concepts of the predictive analysis technique.

An experimental program now in operation for the syntactic analysis of Russian sentences is described in detail in Chapter 5. The aspects of Russian grammar which have been coded in the experimental predictive syntactic analysis program are discussed, and examples are given of both successful and unsuccessful attempts at analysis.

One implication of the model is that a single pass of a sentence through a predictive analysis program does not yield a successful syntactic analysis in all cases. It will be necessary to provide for supplementary passes to correct errors discovered in the initial pass. Many of the errors are easily detected and a scheme for the systematic correction of the errors on subsequent passes seems promising.

When discussing a subject such as syntactic analysis, it is important to distinguish among the <u>use</u>, <u>mention</u>, and <u>representation</u> of a word. Conventionally, a word is <u>used</u> to specify a distinct object, a certain action, etc. But when the word itself is the subject of discussion, its <u>mention</u> facilitates the treatment of the word as an abstract entity, while the <u>representation</u> of a word permits the individual characters to be considered as separate entities. The problem of distinguishing the use, mention, and representation of signs is illustrated by the following examples utilizing Oettinger's convention.[2]

Boston is a city. (Use)

<u>Boston</u>* is an English word. (Mention)

"Boston" is the conventionally spelled representation of <u>Boston</u>*

The asterisk is added to the underscore to denote mention, as distinct from an underscore used alone merely for emphasis.

This notation will be used only when required for the sake of clarity. In Chapters 2 and 3, in particular, it will be used liberally.

REFERENCES

1. Giuliano, V.E., "An Experimental Study of Automatic Language Translation," Doctoral Thesis, Harvard University (1959).

2. Oettinger, A.G., <u>Automatic Language Translation: Lexical and Technical Aspects</u>, Harvard University Press, Cambridge (1960).

CHAPTER 2

AN IDEALIZED CANONICAL STEM DICTIONARY

1. Introduction

In the field of data processing in general, the description of complex systems presents difficult problems. In particular, it has proved difficult to describe with sufficient detail and accuracy the operation of nonnumerical systems. Numerical work can be set forth in mathematical notation, so that it is not necessary to rely on detailed programs to describe the procedures involved. Nonnumerical problems such as automatic translation have similar details, but there is no universal notation for the processes involved or the entities to be manipulated. In general, the procedure has been either to outline processes with flowcharts of increasing complexity, or to give all the details with the operating program itself. However, such a complete description makes the process unintelligible. In particular, this has been the case with automatic translation where it is extremely difficult to design, comprehend, and evaluate such systems.

Recently Iverson has devised a technique of notation that shows some promise of coping with the descriptive problems (Appendix A). One of its striking merits is that it is independent of the characteristics of specific computing machines, and once mastered is of sufficient generality to describe a variety of processes. It seems desirable to formulate a general process of dictionary compilation and operation in terms of this notation.

In this chapter, an idealized canonical dictionary system is presented for the purpose of outlining the essential features of any such system. Besides putting into perspective the essential lexicographic problems of translation, this exposition provides a frame of reference against which the Harvard Automatic Dictionary and other automatic dictionaries can be compared.

A number of basic terms are considered in the following paragraphs. A canonical dictionary is one in which the canonical form of a word, such as the nominative singular of a noun or the infinitive of a verb, is used as the basic source of the dictionary entry (or entries) necessary to represent all the possible inflected forms of the word. In contrast, a dictionary in which the entries are directly generated from text occurrences would not be a canonical dictionary, since any form of a word could occur in a text. Different types of canonical dictionaries are possible. For example, the ordinary dictionary in which the canonical form itself is listed is a canonical dictionary. A second type is a canonical stem dictionary which lists only the stems of the inflected forms, which in turn are obtained from a canonical form. A canonical stem dictionary, to which all further discussion in this chapter will be restricted, is useful only insofar as the number of dictionary entries per word, which averages about ten in a Russian full paradigm dictionary (that is, a dictionary containing every distinct inflected form of a word), can be minimized.

The grammatical attributes of a word can be divided into both lexical attributes and syntactic attributes; the former are determined by examining individual words, while the latter can be determined only by examining the words in context. In a highly inflected language such as Russian, many of the grammatical attributes are lexical, while in a

relatively uninflected language like English, few of the grammatical attributes are lexical and a correspondingly greater number are syntactic. For example, the Russian noun <u>стола</u>* has lexical attributes of case, number, and gender, such that the noun is genitive, singular, and masculine. The English noun <u>table</u>* from the equivalent <u>of the table</u>* has only the lexical attributes of number and gender. The genitive case can be determined only by examining the context in which <u>table</u>* is found.

In the Russian language, the lexical attributes, which are a desired output of a dictionary, are determined by a set of letter combinations called <u>desinences</u> which occur at the ends of words. The desinences cannot be factored systematically as, for example, in the two forms "атом" and "атомом". In the former form the "ом" is part of the stem, while in the latter form the rightmost "ом" is the desinence. It is possible to define an arbitrary set of letter combinations, which will be called <u>affixes</u>, that closely parallel the set of desinences, so that if a word is considered as a string of letters, then the affixes can be factored systematically from the end of the string. The <u>stem</u> is the string of letters which remains after the affix has been removed.

The rest of this chapter is devoted to a discussion of the problems of compilation and operation of a canonical stem dictionary. The problems have been divided into three general areas:

(1) Since it is the set of affixes that is factored in the operation of a stem dictionary, the lexical attributes which are associated with the desinences must be associated with the affixes. In Sec. 2 a scheme is developed for determining the mapping of the desinences onto the affixes in order to associate the lexical attributes with the affixes.

(2) Frequently, the stems of two different words are identical although the set of affixes associated with the individual stems do not intersect at all. A technique by which a list of the affixes associated with a given stem could be stored in the dictionary entry would reduce the nonessential ambiguity in the dictionary file (Sec. 3). Dictionary look-up would be simplified if this technique also provided for the storage of the lexical attributes that are associated with the affixes (as discussed in the previous paragraph).

(3) The look-up process, which has to be repeated for every word looked up in the dictionary file (Sec. 4), should be as simple as possible. A list of the lexical attributes of the text word should be included in the output of the process.

2. Reference Matrices

It is convenient to list the lexical properties, such as case and number for nouns, relevant to the operation of an automatic dictionary before preparing a procedure for compilation or look-up. Since in a Russian stem dictionary the affix of a word is used to determine the lexical properties of the word, the list should consist of all the possible affixes and all the possible lexical attributes. A reference matrix is such a list (Fig. 2-1). One or more reference matrices can be used for an automatic stem dictionary of a given language, depending on the number of attributes and the separability of any of the sets of attributes into disjoint classes.

For a Russian stem dictionary, three productive morphological types t_m (noun, adjective, and verb) have been chosen, and a reference matrix has

Ns	Ns	Ns	As	As	Ip	Ip	Ip	Pp	Pp
#	а	ях	#	а	ами	ьми	ями	ах	ях

Reference Matrix for Noun Morphological Type

Fig. 2-1

been associated with each of these types. Although as many reference matrices as desired may be chosen, a desire for simplicity dictates a search for a natural decomposition of the set of Russian words into several sets of morphological types of words, in order to avoid encountering unnecessary complications, several of which will be illustrated later. Similarly, any arbitrary set of affixes may be used, although the closer the set of affixes parallels the set of desinences, again, the fewer unnecessary complications will be encountered.

A vector $\underline{\lambda}_m$ of lexical attributes associated with a morphological type t_m is defined; for example, if t_m is a noun type: $\underline{\lambda}_m = [\lambda_1, \lambda_2, \ldots, \lambda_{12}]$, where each component of $\underline{\lambda}_m$ is a unique lexical attribute such as nominative singular or genitive plural. The symbols t_m and $\underline{\lambda}_m$ as well as the symbols that will be introduced in succeeding paragraphs are summarized in Table 2-1.

A vector, each of whose components is one of the Russian desinences, and which includes every desinence once and only once, is designated $\underline{\delta}$. Likewise, a vector, each of whose components is a Russian affix factored from a string of letters by an arbitrary algorithm, and which includes every affix once and only once, is designated $\underline{\alpha}$. The order of the components in the vectors $\underline{\delta}$ and $\underline{\alpha}$ is immaterial. The vector $\underline{u}_m(x)$ represents the lexical attributes (there may be more than one) in type t_m of an affix

Symbol	Function
$\underline{\delta}$	Desinence vector
$\underline{\alpha}$	Affix vector
t_m	Morphological type
$\underline{\lambda}_m$	Lexical attributes of morphological type t_m
$\underline{\mu}_m(x)$	Lexical attributes of desinence or affix x
$\underline{V}$	Reference matrix V_i^1 - lexical attribute V_i^2 - affix
$\underline{V}^*$	Auxiliary reference matrix V_i^{*1} - lexical attribute V_i^{*2} - desinence
$F(x)$	Arbitrary factoring operation on word x
$\underline{\Pi}(x)$	Paradigm representation of word x
$\underline{\gamma}_k$	Entry function vector
α_w	Affix of word w
$\underline{\lambda}_w$	Lexical attributes of word w
i,j,k,	Indices
Λ	Null formula

Definition of Symbols

TABLE 2-1

or desinence x, thus, where "ом" and "а" are desinences, $\underline{u}_{noun}$ ("ом") = [instrumental singular], while $\underline{u}_{noun}$ ("а") = [nominative singular, genitive singular, accusative singular, nominative plural, accusative plural].

All the information known about a morphological type prior to the construction of a reference matrix can be summarized in the list of lexical attributes, the list of all possible affixes, the list of all possible desinences, and the set of vectors $\underline{u}_m(\delta_i)$. However, since affixes and not desinences are factored from words, a condensed representation of the set of vectors $\underline{u}_m(\alpha_i)$ is needed. The rest of this section is devoted to the problem of obtaining this condensed representation.

For each lexical attribute of a given morphological type a two-row submatrix is constructed such that the components in the second row represent affixes that can signify the lexical attribute. Each component in the first row represents the lexical attribute itself (Fig. 2-2).

Pp	Pp
ах	ях

Submatrix of the Lexical Attribute Prepositional
Plural of the Noun Morphological Type

Fig. 2-2

The submatrices of all the lexical attributes are then joined to form the reference matrix (Fig. 2-1). The ordering of the submatrices must coincide with the ordering of the lexical attributes in $\underline{\lambda}_m$, but the ordering of the columns within each submatrix is immaterial. Each affix can occur no more than once in a submatrix, but can be repeated in any number of submatrices.

The operations necessary to construct a reference matrix from the affix vector, the desinence vector, the lexical attribute vector $\underline{\lambda}_m$, and the set of vectors $\underline{u}_m(\delta_i)$ are shown in Program 2-1 and explained in the following paragraphs.

Prior to constructing the reference matrix $\underline{V}$, it is convenient to construct an auxiliary matrix $\underline{V}^*$ that resembles the reference matrix in form but whose second row is a row of desinences instead of affixes. The matrix $\underline{V}^*$ is set to null in step 1.

To iterate over all the lexical attributes in $\underline{\lambda}_m$, an index i is initialized in step 2 and decremented in step 3. A minor loop for each desinence is initialized in step 4. Step 5 sets the logical vector $\underline{p}$ to null, and step 6 decrements the index j of the minor loop.

In step 7, the component λ_{m_i} of $\underline{\lambda}_m$ is treated as a vector of one component.* This component is mapped onto $\underline{u}_m(\delta_j)$, the lexical attributes of the component δ_j of $\underline{\delta}$. Since $\underline{\lambda}_{m_i}$ has but one component, the resultant of the mapping is an integer. The logical reduction substitutes a "1" in the place of any integer other than "0", and the "1" or the "0" is left-adjoined to $\underline{p}$. This iterative process is repeated until every component of $\underline{\delta}$ has been scanned.

For example, if $\lambda_{m_i} = \left[\text{nom. sing.}\right]$ and δ_j = "a", then $\underline{u}_m(\delta_j) =$ [nom. sing., gen. sing., accus. sing., nom. plur., accus. plur.], $\mu\left[\underline{u}_m(\delta_j) \longleftarrow \underline{\lambda}_{m_i}\right] = 1$, and since $1 \neq 0$, a "1" is left-adjoined to $\underline{p}$. If

*Throughout this and succeeding programs, a string of characters will be considered both as a one-component vector and as a vector with each character of the string a component of a vector. Thus, $\underline{u}$ = ["green"] is a one-component vector, but $\underline{v}$ = ["g", "r", "e", "e", "n"] is a five-component vector. It is also possible that the entire string will be but one component of a vector in another context; for example, the three-component vector $\underline{w}$ = ["the", "green", "leaf"].

1 $\underline{V}^{*} \leftarrow \Lambda$

2 $i \leftarrow \nu(\underline{\lambda}_m)\uparrow$

3 $i \leftarrow i\downarrow$

4 $j \leftarrow \nu(\underline{\delta})\uparrow$

5 $\underline{p} \leftarrow \Lambda$

6 $j \leftarrow j\downarrow$

7 $\underline{p} \leftarrow \{\mu[\underline{u}_m(\delta_j) \leftarrow \underline{\lambda}_{m_i}] \neq 0\}, \underline{p}$

8 $\underline{V}^{*2} \leftarrow \underline{p}/\underline{\delta}, \underline{V}^{*2}$

9 $\underline{V}^{*1} \leftarrow \lambda_{m_i}\underline{\epsilon}, \underline{V}^{*1}$

10 $\underline{V} \leftarrow \Lambda$

11 $k \leftarrow \nu(\underline{V}^{*})\uparrow$

12 $k \leftarrow k\downarrow$

13 $l \leftarrow \nu(\underline{a})\uparrow$

14 $\underline{\beta} \leftarrow \Lambda$

15 $l \leftarrow l\downarrow$

16 $\underline{\beta} \leftarrow [t^{\nu(V^{*2}_k)}/\underline{a}_l], \underline{\beta}$

17 $\underline{\gamma} \leftarrow (V^{*2}_k\,\underline{\epsilon} = \underline{\beta})/\underline{a}$

18 $\nu(\gamma) \quad : \quad 0$

19 $\gamma^{*2}_k \leftarrow F(\underline{V}^{*2}_k)/\underline{V}^{*2}_k$

20 $\underline{V}^{2} \leftarrow \underline{\gamma}, \underline{V}^{2}$

21 $\underline{V}^{1} \leftarrow V^{*1}_k\,\underline{\epsilon}, \underline{V}^{1}$

Program for Constructing a Reference Matrix for a Morphological Type

Program 2-1

$\lambda_{m_i} = \left[\text{nom. plur.}\right]$ for the same δ_j, then $\mu\left[\underline{u}_m(\delta_j) \leftarrow \underline{\lambda}_{m_i}\right] = 4$, but since $4 \neq 0$, a "1" would still be left-adjoined to $\underline{p}$. If δ_j = "ом", then $\underline{u}_m(\delta_j) = \left[\text{instr. sing.}\right]$, $\mu\left[\underline{u}_m(\delta_j) \leftarrow \underline{\lambda}_{m_i}\right] = 0$, and a "0" is left-adjoined to $\underline{p}$.

The resultant logical vector $\underline{p}$ of this iterative loop is of the dimension of $\underline{\delta}$ and has a "1" in each location corresponding to the components of $\underline{\delta}$ which could have the lexical attribute λ_{m_i} in t_m.

In step 8, $\underline{\delta}$ is compressed by $\underline{p}$. The compressed subvector of $\underline{\delta}$ is left-adjoined to the second row of $\underline{V}^*$. The components of the subvector are the desinences that have the lexical attributes λ_{m_i} in t_m, for instance: the subvector for the lexical attribute prepositional plural in the noun type would be $\left[\text{ax}, \text{яx}\right]$ (Fig. 2-2). A vector, each of whose components is λ_{m_i}, and of the dimension of the desinence subvector, is left-adjoined to the first row of $\underline{V}^*$ in step 9.

This entire process is repeated until a submatrix has been adjoined to $\underline{V}^*$ for every lexical attribute in $\underline{\lambda}_T$. In the next sequence of steps, each desinence (column) in $\underline{V}^*$ is replaced by a submatrix of affixes in $\underline{V}$. Any of these affixes might be factored from a string of letters ending in the desinence; for example, the affixes "у", "ему", or "ому" might be factored from a string of letters ending in the desinence "у".[†] The arbitrary factoring operation used in this process is designated F(x), where x is the string of characters being factored, and a logical tail vector $\underline{q}$ is defined as the result of the operation F on the string x, $\underline{q} = F(x)$. The

[†] The factoring of the string "ему" or "ому" is an example of false factoring if the desinence is in fact "у" (Sec. 3.3C).

weight of the logical tail vector $\underline{q}$ is equal to the dimension of the affix factored by F, and the dimension of $\underline{q}$ is equal to the dimension of the original string of letters x.

The reference matrix is set to null in step 10. An iterative process that will operate on each column of $\underline{V}^*$ is initialized in step 11 and the index k is decremented in step 12.

A minor loop to scan $\underline{\alpha}$ for each $\underline{V}^*_k$ is initialized in step 13. The vector $\underline{\beta}$ is set to null in step 14 and the index ℓ is decremented in step 15.

In step 16, the affix α_ℓ in $\underline{\alpha}$ is considered a vector with the individual letters of the affix as components of the vector. This vector is compressed by a logical tail vector whose weight is equal to the dimension of the desinence V^{*2}_k, which is also considered a vector with letters as components in this process. This step ensures compatibility between the elements of $\underline{\beta}$ and V^{*2}_k in the next step. Thus, if V^{*2}_k = "y" and if α_ℓ = "emy", then $\nu(\underline{V}^{*2}_k) = 1$, $t^{\nu(\underline{V}^{*2}_k)} = [0\ldots01]$, and $t^{\nu(\underline{V}^{*2}_k)}/\alpha_\ell$ = "y". The element adjoined to $\underline{\beta}$ has the same dimension as $\underline{V}^{*2}_k$. It should be noted that by the definition of a logical tail vector, if $\sigma(t) > \nu(\alpha_\ell)$, then α_ℓ remains unchanged, as in the case if V^{*2}_k = "emy" and α_ℓ = "y" where $t^{\nu(\underline{V}^{*2}_k)}/\alpha_\ell$ = "y".

In step 17, the components of $\underline{\beta}$ are logically reduced by a vector each of whose components is V^{*2}_k. The resultant logical vector is used to compress $\underline{\alpha}$. The compressed vector $\underline{\gamma}$ contains, as components, all the affixes that might be factored by the arbitrary factoring algorithm operating on a string of letters ending in the desinence V^{*2}_k.

If the dimension of $\underline{\gamma}$ is zero (step 18), then no affix that is at least as long as the desinence represented by V^{*2}_k exists; and the desinence can be replaced only by an affix shorter than the desinence. The desinence is factored by F in step 19 and the resulting affix is substituted for the desinence in V^{*2}_k, after which the process returns to step 13. This path can be followed only once per V^{*2}_k, since $\nu(\gamma) \neq 0$ in step 18 once the affix has been substituted for the desinence.

If the dimension of $\underline{\gamma}$ is not zero at step 18, then one or more affix, that might be factored by the algorithm when operating on a word ending in the desinence, has been found. The program transfers to step 20 and $\underline{\gamma}$ is left-adjoined to $\underline{V}^2$. In step 21, a vector, each of whose components is V^{*1}_k, and of the same dimension as $\underline{\gamma}$, is left-adjoined to $\underline{V}^1$.

The process of steps 12 to 21 is repeated for every desinence in $\underline{V}^*$ until the reference matrix $\underline{V}$ has been completely generated.

As an illustration of the entire process of producing a reference matrix, a greatly simplified morphological type with only three lexical attributes, dative singular (Ds), prepositional singular (Ps), and instrumental plural (Ip), will be considered. The range of desinences corresponding to these morphological types will also be limited.

The step-by-step process is outlined in detail in Table 2-2 based on the following set of definitions:

$$\underline{\delta} = \left[\text{ам},\ \text{ами},\ \text{е},\ \text{и}\right],$$

$$\underline{\alpha} = \left[\text{ам},\ \text{ами},\ \text{е},\ \text{ие},\ \text{и}\right],$$

$$\underline{\lambda}_m = \left[\text{Ds},\ \text{Ps},\ \text{Ip}\right];$$

Step	Other Conditions	Result
2		$i = 4$
4		$j = 5$
7	$i = 3,\ j = 4$	$([\mathrm{Ps}] \longleftarrow [\mathrm{Ip}]) = 0,\ \underline{p} = [0]$
7	$i = 3,\ j = 3$	$([\mathrm{Ds},\mathrm{Ps}] \longleftarrow [\mathrm{Ip}]) = 0,\ \underline{p} = [0,0]$
7	$i = 3,\ j = 2$	$([\mathrm{Ip}] \longleftarrow [\mathrm{Ip}]) = 1,\ \underline{p} = [1,0,0]$
7	$i = 3,\ j = 1$	$(\Lambda \longleftarrow [\mathrm{Ip}]) = 0,\ \underline{p} = [0,1,0,0]$
8	$i = 3$	$\underline{V}^{*2} = [\text{ами}]$
9		$\underline{V}^{*1} = [\mathrm{Ip}]$
7	$i = 2,\ j = 1$	$\underline{p} = [0,0,1,1]$
9	$i = 2$	$\underline{V}^{*} = \begin{bmatrix} \mathrm{Ps} & \mathrm{Ps} & \mathrm{Ip} \\ \text{е} & \text{и} & \text{ами} \end{bmatrix}$
7	$i = 1,\ j = 1$	$\underline{p} = [0,0,1,0]$
9	$i = 1$	$\underline{V}^{*} = \begin{bmatrix} \mathrm{Ds} & \mathrm{Ps} & \mathrm{Ps} & \mathrm{Ip} \\ \text{е} & \text{е} & \text{и} & \text{ами} \end{bmatrix}$
11		$k = 5$
13		$\ell = 6$
16	$k = 4,\ \ell = 1$	$\underline{\beta} = [\text{ам, ами, е, ие, и}]$
17	$k = 4$	$(\underline{V}^{*2}_{k} = \underline{\beta}) = [0,1,0,0,0],\ \underline{V}^{2} = [\text{ами}]$
18		$\underline{V}^{1} = [\mathrm{Ip}]$
16	$k = 3,\ \ell = 1$	$\underline{\beta} = [\text{м, и, е, е, и}]$
18	$k = 3$	$(\underline{V}^{*2}_{k} = \underline{\beta}) = [0,1,0,0,1]$
		$\underline{V} = \begin{bmatrix} \mathrm{Ps} & \mathrm{Ps} & \mathrm{Ip} \\ \text{ами} & \text{и} & \text{ами} \end{bmatrix}$
16	$k = 2,\ \ell = 1$	$\underline{\beta} = [\text{м, и, е, е, и}]$
18	$k = 2$	$(\underline{V}^{*2}_{k} = \underline{\beta}) = [0,0,1,1,0]$
		$\underline{V} = \begin{bmatrix} \mathrm{Ps} & \mathrm{Ps} & \mathrm{Ps} & \mathrm{Ps} & \mathrm{Ip} \\ \text{е} & \text{ие} & \text{ами} & \text{и} & \text{ами} \end{bmatrix}$
18	$k = 1$	$\underline{V} = \begin{bmatrix} \mathrm{Ds} & \mathrm{Ds} & \mathrm{Ps} & \mathrm{Ps} & \mathrm{Ps} & \mathrm{Ps} & \mathrm{Ip} \\ \text{е} & \text{ие} & \text{е} & \text{ие} & \text{ами} & \text{и} & \text{ами} \end{bmatrix}$

Details in the Process of Producing a Reference Matrix

TABLE 2-2

and for each desinence,

$$u_m(\text{ам}) = \Lambda\,,$$
$$u_m(\text{ами}) = [\mathrm{Ip}],$$
$$u_m(\text{е}) = [\mathrm{Ds},\ \mathrm{Ps}],$$
$$u_m(\text{и}) = [\mathrm{Ps}].$$

3. Dictionary Compilation

Every stem of an inflected word is stored as a separate dictionary entry. Every dictionary entry will contain a list of the set of affixes that can occur with the stem and the lexical attributes associated with each affix. This information will be represented by a logical vector, the <u>entry function vector</u> $\underline{y}$, such that $\nu(\underline{y}) = \nu(\underline{V})$. Every "1" in the entry function vector will correspond to the affix - lexical attribute pair of the corresponding column of the reference matrix. For instance, if a stem in the morphological class of Table 2-2 had the affix "е" in the dative singular, "и" in the prepositional singular, and "ами" in the instrumental plural, then the entry function vector of that stem $\underline{y} = [1,0,0,0,0,1,1]$.

The compilation of a set of entry function vectors $\underline{y}_k$ (there are k stems in the paradigm of a word) will now be considered. The reference matrix, the paradigm of the word, and an arbitrary factoring algorithm are necessary initially for the compilation.

The paradigm representation of an inflected word w, belonging to class c_i in morphological type t_m is denoted by the matrix $\underline{\Pi}$ where $\nu(\underline{\Pi}) = 2$ (Fig. 2-3). All the relevant lexical attributes are listed in the first column and all the members of the paradigm of the word are listed in the

second column of the matrix. Each row consists of a single member of the paradigm and its associated lexical attribute.

The process for obtaining the entry function vectors for a paradigm is described in Program 2-2.

$$\underline{\Pi}\,(\underline{\text{атом}}^{*}) = \begin{bmatrix} \text{Ns} & \text{атом} \\ \text{Gs} & \text{атома} \\ \text{As} & \text{атом} \\ \text{Ds} & \text{атому} \\ \text{Is} & \text{атомом} \\ \text{Ps} & \text{атоме} \\ \text{Np} & \text{атомы} \\ \text{Gp} & \text{атомов} \\ \text{Ap} & \text{атомы} \\ \text{Dp} & \text{атомам} \\ \text{Ip} & \text{атомами} \\ \text{Pp} & \text{атомах} \end{bmatrix}$$

Paradigm Representation Matrix for $\underline{\text{атом}}^{*}$

Fig. 2-3

Step 1 defines the paradigm representation, $\underline{\Pi}$, according to the rules of the class c_i to which the word w belongs. In step 2, the arbitrary factoring algorithm F is applied to the second component of each row of $\underline{\Pi}$, each of these components being considered a vector. The resulting logical tail vector is the corresponding component of the column vector $\underline{\omega}$. Figure 2-4 shows $\underline{\omega}$ for the paradigm representation of Fig. 2-3, using an arbitrary algorithm that factors the affixes: "ом", "а", "ому", "е", "ы", "ов", "ам", "ами", and "ах", among others.

The second component of each row of $\underline{\Pi}$ is compressed by the inverse of the corresponding component of $\underline{\omega}$, and the resultant components of $\underline{\Pi}$ are components of the vector $\underline{\sigma}$ in step 3. Each component is a stem from the paradigm representation $\underline{\Pi}$; thus, using the same example,

$\underline{\sigma}$ = [ат, атом, ат, ат, атом, атом, атом, атом, атом, атом, атом, атом].

1 $\underline{\Pi} \leftarrow \Pi\left[c_i(w)\right]$

2 $\underline{\omega}^{\epsilon} \leftarrow F(t^{1}/\underline{\Pi}^{\epsilon})$

3 $\underline{\sigma}_{\epsilon} \leftarrow \left[\overline{\underline{\omega}}^{\epsilon}/(\underline{t}^{1}/\underline{\Pi}^{\epsilon})\right]$

4 $\underline{\rho} \leftarrow \left[\underline{\nu} = \mu(\underline{\sigma} \leftarrow \underline{\sigma})\right]/\underline{\sigma}$

5 $k \leftarrow \nu(\rho)\uparrow$

6 $k \leftarrow k\downarrow$ $=$

7 $\underline{\Psi}_{k} \leftarrow \Lambda$

8 $\underline{\Psi}_{\epsilon} \leftarrow \left[(\underline{\rho}_{k}{}^{\epsilon} = \underline{\sigma}) /\!/ \underline{\Pi}\right]^{\epsilon}$

9 $j \leftarrow \nu(\underline{\Psi})\uparrow$

10 $=$ $j \leftarrow j\downarrow$

11 $\Psi_{j}^{2} \leftarrow F(\underline{\Psi}_{j}^{2}) / \underline{\Psi}_{j}^{2}$

12 $\underline{\Psi}_{k} \leftarrow \left[\mu(\underline{\Psi}_{\epsilon} \leftarrow \underline{\Psi}_{\epsilon}) \neq 0\right]$

Program for Constructing Entry Function Vectors
for an Inflected Word

Program 2-2

$$\underline{\omega} = F\left(\underline{\Pi}\left[\underline{\text{atom}^{*}}\right]\right) = \begin{bmatrix} 0011 \\ 00001 \\ 0011 \\ 00111 \\ 000011 \\ 00001 \\ 00001 \\ 000011 \\ 00001 \\ 000011 \\ 0000111 \\ 000011 \end{bmatrix}$$

Logical Column Vector Resulting from the Factoring
of the Paradigm Representation of $\underline{\text{atom}^{*}}$

Fig. 2-4

The vector $\underline{\sigma}$ is mapped onto itself in step 4, resulting in a permutation vector which, in turn, is compared with the identity permutation vector $\underline{\nu}$; in the case of $\underline{\text{атом}}^*$, the permutation vector $\underline{n} = (1,2,1,1,2,2,2,2,2,2,2,2)$ is obtained. The resultant logical vector is then used to compress $\underline{\sigma}$. The consequence of this operation is to determine the vector $\underline{\rho}$ derived from $\underline{\sigma}$ by suppressing repeated components, such that each distinct stem of $\underline{\rho}$ is a component of $\underline{\sigma}$, thus for $\underline{\text{атом}}^*$, $\underline{\rho} = \left[\text{ат, атом}\right]$.

The index k is initialized in step 5 and decremented in step 6 for the iterative process that will create k dictionary entries from the paradigm representation of w.

In step 7, the vector $\underline{y}_k$ which will be the entry function vector for the stem ρ_k, is set to all zeros. The dimension of $\underline{y}_k$ is the same as the row dimension of the reference matrix for the type t_m of the word under consideration.

In step 8, the components of $\underline{\sigma}$ are logically reduced by a vector, each component of which is ρ_k. The columns of $\underline{\Pi}$ are compressed by the resultant logical vector, each remaining row of $\underline{\Pi}$ becoming a column of Ψ. The resultant subparadigm representation $\underline{\Psi}$ of w contains all the inflected forms of the paradigm representation that result in the stem ρ_k after being factored by the arbitrary algorithm.

Once more considering the paradigm representation of $\underline{\text{атом}}^*$, for the stem "ат",

$$\Psi(\text{ат}) = \begin{bmatrix} \text{Ns} & \text{As} & \text{Ds} \\ \text{атом} & \text{атом} & \text{атому} \end{bmatrix},$$

while for the stem "атом",

$$\underline{\Psi}(\text{атом}) = \begin{bmatrix} Gs & Is & Ps & Np & Gp & Ap & Dp & Ip & Pp \\ \text{атома} & \text{атомом} & \text{атоме} & \text{атомы} & \text{атомов} & \text{атомы} & \text{атомам} & \text{атомами} & \text{атомах} \end{bmatrix}$$

In step 9, the index j is initialized and then decremented in step 10 for an iterative process on every component of the second row of the subparadigm representation matrix $\underline{\Psi}$.

In step 11, the arbitrary factoring algorithm operates on the inflected form Ψ_j^2, which is regarded as a vector with letters as components. The resultant logical vector then is used to compress Ψ_j^2, resulting in the replacement of the inflected form by its affix. This process is repeated for every inflected form in $\underline{\Psi}^2$, such that, in the example,

$$\underline{\Psi}(\text{ат}) = \begin{bmatrix} Ns & As & Ds \\ \text{ом} & \text{ом} & \text{ому} \end{bmatrix},$$

and

$$\underline{\Psi}(\text{атом}) = \begin{bmatrix} Gs & Is & Ps & Np & Gp & Ap & Dp & Ip & Pp \\ \text{а} & \text{ом} & \text{е} & \text{ы} & \text{ов} & \text{ы} & \text{ам} & \text{ами} & \text{ах} \end{bmatrix}$$

In step 12, every column of the reference matrix $\underline{V}$ is mapped onto the columns of the subparadigm representation matrix $\underline{\Psi}$. Thus for every column in $\underline{V}$ that also exists in $\underline{\Psi}$ there will be a "1" in the corresponding element of the logical vector $\underline{y}_k$.

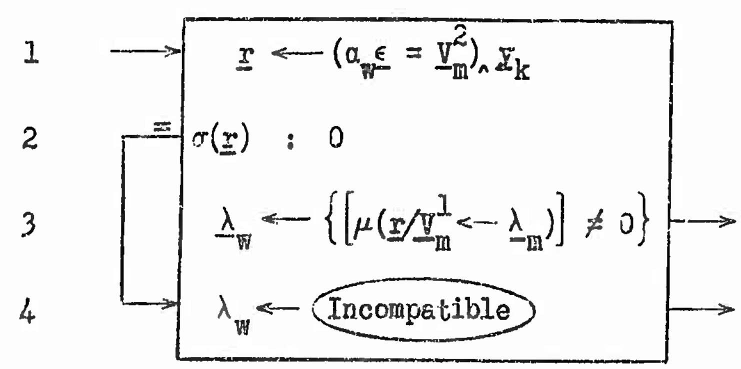
A technique for storing a mixed canonical stem and full paradigm dictionary is suggested by the entry function vector. If, for a given word, a mark were entered in some extra register to indicate that the word is to be stored as a full paradigm, then the step $\underline{\sigma}_\epsilon \longleftarrow \underline{t}^1/\underline{\Pi}^\epsilon$ could be substituted for steps 2 and 3 of Program 2-2 and an entry for every distinct inflected form of a paradigm would be generated. With this technique, the dictionary look-up process which will be described in the next section would not have

to be altered at all to look up words in the mixed stem and full paradigm dictionary.

4. Analysis of Inflected Words

An entry in the idealized dictionary can be looked up by using the stem of the word as the key. Once a dictionary entry has been found, it is necessary to determine whether the affix factored from the text word can occur legitimately with the stem of the dictionary entry. If so, the lexical attributes (there may be more than one) of that affix are displayed in a condensed logical vector, the <u>reduced lexical attribute vector</u>, of the dimension of $\underline{\lambda}_m$.

Once a dictionary entry has been found, it is necessary only to compare the affix of the text word with the list of all possible affixes that is stored in the form of the entry function vector (Program 2-3). If the affix of the text word corresponds with one or more affixes on the list, the corresponding lexical attributes are displayed.



Program for Determining Compatibility of Dictionary Entry
and Compressing Lexical Attributes for Text Word

Program 2-3

The whole second row of the reference matrix $\underline{V}_m$ of morphological type t_m is logically reduced in step 1 by a vector each component of which is the affix α_w. The resulting logical vector is intersected with the entry function vector $\underline{y}_k$ which is stored in the dictionary entry. If the resultant logical vector has only zero components, the dictionary entry is <u>incompatible</u> with the text word, that is, the word represented by the dictionary entry is not the same word as the word encountered in the text.

If the dictionary entry is <u>compatible</u>, the first row of the reference matrix is compressed by the logical vector $\underline{r}$ in step 2. The lexical attribute vector $\underline{\lambda}_m$ is then mapped onto the compressed row of $\underline{V}$. The resultant logical vector is the reduced lexical attribute vector, $\underline{\lambda}_w$.

As an example, the simplified reference matrix of Sec. 2 and the subsequent entry function vector of Sec. 3 will be used:

$$\underline{V} = \begin{bmatrix} \text{Ds} & \text{Ds} & \text{Ps} & \text{Ps} & \text{Ps} & \text{Ps} & \text{Ip} \\ \text{е} & \text{ие} & \text{е} & \text{ие} & \text{ами} & \text{и} & \text{ами} \end{bmatrix},$$

$$\underline{y}_k = \left[1,0,0,0,0,1,1\right].$$

If the affix "ие" is factored from a text word with a stem that results in the look-up of the entry with the entry function vector $\underline{y}_k$, then the logical reduction $(\underline{\alpha_w \epsilon} = \underline{V}_m^2) = \left[0,1,0,1,0,0,0\right]$ and the intersection will result in $\underline{V} = \left[0,0,0,0,0,0,0\right]$ and the subsequent interpretation that the affix is incompatible with the stem of that dictionary entry. However, if the affix "ами" is factored, then $(\underline{\alpha_w \epsilon} = \underline{V}_m^2) = \left[0,0,0,0,1,0,1\right]$, $\underline{r} = \left[0,0,0,0,0,0,1\right]$, after which $\underline{r}/\underline{V}_m^1 = \left[\text{Ip}\right]$, $\mu\left[\underline{r}/\underline{V}_m^1 \leftarrow \underline{\lambda}_m\right] = \left[0,0,1\right]$, $\underline{\lambda}_w = \left[0,0,1\right]$, and the stem of the dictionary entry is compatible with the affix "ами".

5. Summary

The three programs described in this chapter constitute necessary steps for the compilation and operation of an idealized stem dictionary. Keeping in mind some constraints of practical data processing, the most complex set of instructions has been used for the creation of the reference matrices, a task that has to be performed relatively few times, while the simplest set of instructions has been used in the operation of the dictionary, the task that has to be performed most frequently.

Although the dictionary described above is idealized, it is highly impractical. The necessary operations for dictionary compilation and for the analysis of dictionary entries are well defined, but too many machine words are necessary to store the reference matrices and the entry function vectors even on a binary machine where each bit is individually accessible. Many more than 100 bits are needed for each entry function vector, since a desinence often has more than one lexical attribute, each of which is represented by a bit in $\underline{V}^*$, and the desinence is often replaced by several affixes. To operate a practical stem dictionary, it is necessary to avoid using so much storage for each dictionary entry. In the next chapter, where the Harvard Automatic Dictionary will be described, several methods for reducing the storage requirements will be pointed out.

CHAPTER 3

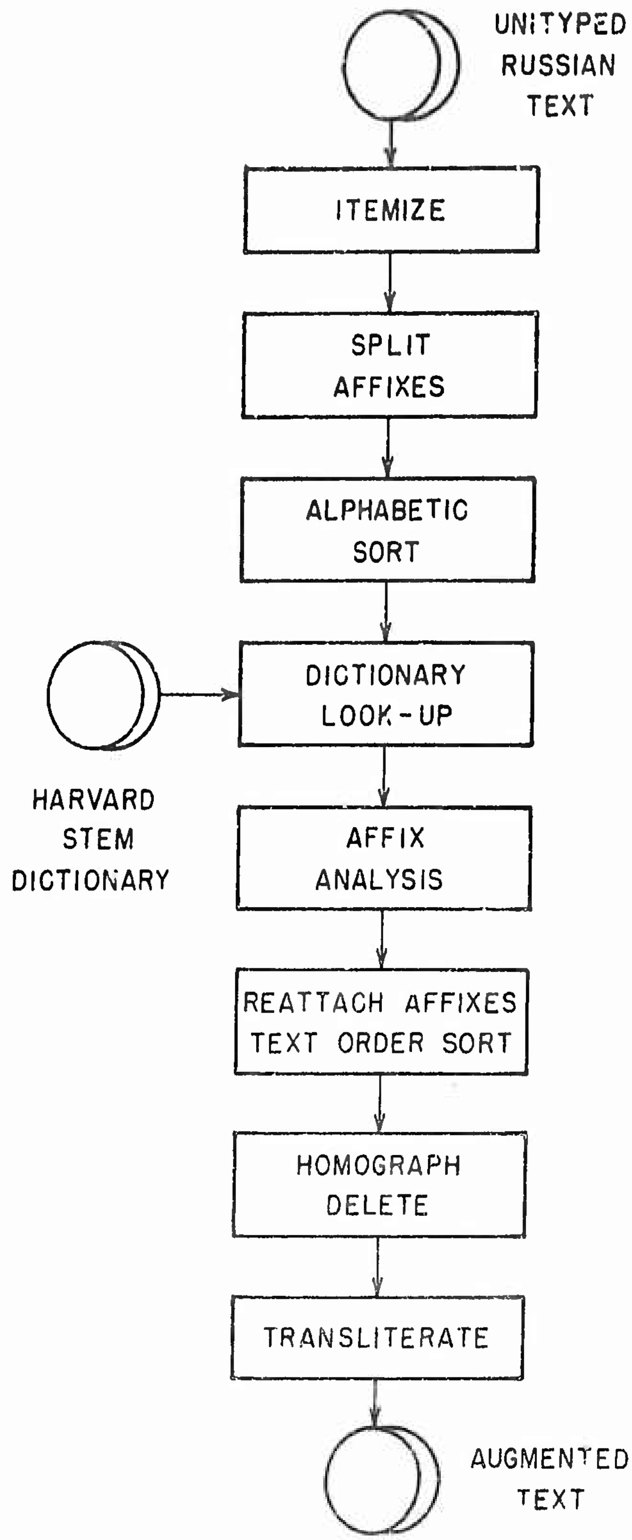
THE HARVARD AUTOMATIC DICTIONARY – AN OPERATING CANONICAL STEM DICTIONARY

1. Introduction

An automatic dictionary is an essential component of an automatic translator. A canonical stem dictionary, the Russian-to-English Harvard Automatic Dictionary, has been put into operation over the last four years and is controlled by a comprehensive set of programs and routines. Giuliano[1] and others[2,3,4] have described the solutions to many of the problems related both to the compilation and modification of the dictionary file and to the look-up of words in the dictionary. The solutions to the remaining problems of word-by-word analysis are considered in this chapter.

The look-up of words† is effected by the Continuous Dictionary Run, a set of programs which are executed continuously and in sequence (Fig. 3-1). A Russian text is copied onto a magnetic tape in a format similar to the original copy. The itemize program organizes the format of the input text, placing each text word into an item of standard size. The affixes are removed from the Russian words by the "inverse inflection algorithm" in the split program, and the items are sorted into alphabetic order with the remaining stem of the word as the primary key. Each stem is then looked up in the dictionary and a complete dictionary entry is substituted for every word in the text. At this point, each word is analyzed morphologically and the syntactic information thus obtained is inserted into the dictionary item.

† Included in the definition of a text word or a word of a sentence will be any punctuation mark, mathematical symbol, abbreviation, etc.



Continuous Dictionary Run

Fig. 3-1

In the next program the affixes are reattached to the stems and the words are sorted back into text order. Following this, "homographs" are deleted, and in the last program the Russian words are transliterated to permit their representation by Latin letters. The output of the Continuous Dictionary Run is referred to as the "augmented text". The programs of the Continuous Dictionary Run have been described by Jones.[5,6]

The classification scheme of Russian words and the inverse inflection algorithm, both of which were developed more than three years ago, are discussed in the light of the experience gained in working with them since that time (Sec. 2). A mapping operation to correlate the classification scheme with the inverse inflection algorithm is presented in Sec. 3.

The system devised to interpret false factoring (that is, the factoring of a string of letters different from the expected affix) of dictionary items by the inverse inflection algorithm (Sec. 4) and the system devised to analyze the affixes of text words given their dictionary entries (Sec. 5) are described in detail. An example of the output of the corresponding programs is presented in Sec. 6.

Some statistics on the reliability of the Harvard Automatic Dictionary are set forth in Sec. 7, while additional statistics for the efficient operation of the analyzing programs are introduced in Sec. 8.

2. Word Classification and the Inverse Inflection Algorithm

The output of the Continuous Dictionary Run contains basically the same grammatical information as the output of the idealized dictionary; however, the mode of operation of the program differs greatly from that of

the idealized system. The various routines that constitute the dictionary compilation system and the Continuous Dictionary Run were written over a time span of several years. In the more recent routines the possibility of using new symbols and formats was often limited by those already adopted during earlier periods of research. This has had strong effects on the mode of operation selected in these newer routines and has imposed many apparently arbitrary constraints on the actual experimental system.

Some of the earlier phases of dictionary research are discussed with the aim of describing them in terms of the idealized dictionary and of pointing out changes that might be made should it become desirable to reprogram the system.

A. Morphological Types and Their Classification

Before the description of the morphological types, Oettinger's definition and notation for paradigms,[7] which will be used in this chapter, is given. A paradigm of a word is the full set of inflected forms of the word. Usually there are twelve inflected forms in noun paradigms (Fig. 3-2). A reduced paradigm of a word is the set of distinct representations (Fig. 3-3). Examination of Fig. 3-2 and Fig. 3-3 points out that there is only one distinct representation "студента" for студента^*_{As}, and студента^*_{Gs}, and one distinct representation "студентов" for студентов^*_{Ap} and студентов^*_{Gp}. This multiple usage of distinct representations defines internal homography. A detailed description of the different types of homography can be found in Chap. 9 of Ref. 7.

студент* Ns	студенты* Np
студента* As	студентов* Ap
студента* Gs	студентов* Gp
студенту* Ds	студентам* Dp
студентом* Is	студентами* Ip
студенте* Ps	студентах* Pp

Paradigm of студент*
Fig. 3-2

"студент"	"студенты"
"студента"	"студентов"
"студенту"	"студентам"
"студентом"	"студентами"
"студенте"	"студентах"

Reduced Paradigm of студент*
Fig. 3-3

In the Harvard Automatic Dictionary inflected words have been arbitrarily divided into three morphological types: nouns, adjectives, and verbs. Each of these types was divided into a number of morphological classes by Magassy.[8,9] The morphological classes were kept as few in number as possible to ease the burden of assigning new words to these classes and to simplify the programs for inflecting these classes during the generation of dictionary entries. Because of this morphological description, it is possible to find some nouns such as портной* belonging to an adjectival morphological class.

In the classification scheme for every noun, adjective, and verb class two important types of information are given (Fig. 3-4). The first is a morphological description of the words that belong to the particular class, stressing the behavior of the word "tails" that can occur. In the example shown, the class N2 consists of all nouns ending in "ой", "ай", and "ий", as well as of some of the nouns ending in "ей". Secondly, for each

CLASS	EXAMPLES	CLASS IDENTIFICATION
N2	строй лишай гений музей	A class embracing: 1. the nouns ending in о, а, и } + й. 2. some nouns ending in е

GENERATION RULES	
Generating Stem (GS)	Generated Forms with Specified Generating Affixes
word − last letter.	a. word b. GS + я c. + ю d. + ем e. + е f. GS + и g. + ев h. + ям i. + ями j. + ях

Definition and Description of Class N2

Fig. 3-4

class there is a generation rule specifying both how the generating stem is formed (in the example, the word less the last letter) and which generating affixes can be right-adjoined to the generating stem to form the members of the reduced paradigm of the word. The generating stem of a noun in class N2 can end in "о", "а", "е", or "и", and, in addition, can have the generating affixes "й", "я", "ю", "ем", "е", "и", "ев", "ям", "ями", and "ях" adjoined. This list of generating affixes completes the description of this class of noun.

Three types of word endings now have been introduced in this thesis. A desinence is a word ending that has lexical significance but which cannot be identified formally. An affix is a word ending approximating a desinence that is factored formally from a word by an appropriate algorithm. A

generating affix is an artificial word ending that is used to construct the reduced paradigm of a word from both its canonical form and its class marker.

All the generating affixes of a class might not be needed to define a single paradigm. Several related paradigms are sometimes fused into a single class, or cumulative paradigm, as an alternative to maintaining separate classes for dictionary compilation; thus, in class N4 the instrumental singular is "дамой" or "улицей", but not "дамей" or "улицой".

Matejka[10,11] eliminated (1) some ambiguities that had been deliberately left in the morphological description of the grammatical functions, and (2) the spurious forms, by separating the noun classes into finer subdivisions. Every noun class was divided into animate and inanimate categories, and these groups were further divided into as many as four categories, although rarely into more than two.

For example, if nouns of class N1, such as студент* (animate) (Figs. 3-2 and 3-3) and стол* (inanimate) (Figs. 3-5 and 3-6) were not subdivided into animate and inanimate categories, paradigmatic forms ending in "#"† would all be either nominative singular or accusative singular, and forms ending in "a" would all be either accusative singular or genitive singular. The identification for студент and стол would be:

"студент" would represent: $\underline{\text{студент}}^{*}_{Ns}$ and $\underline{\text{студент}}^{*}_{As}$,

"студента" would represent: $\underline{\text{студента}}^{*}_{As}$ and $\underline{\text{студента}}^{*}_{Gs}$,

"стол" would represent: $\underline{\text{стол}}^{*}_{Ns}$ and $\underline{\text{стол}}^{*}_{As}$,

"стола" would represent: $\underline{\text{стола}}^{*}_{As}$ and $\underline{\text{стола}}^{*}_{Gs}$.

† The symbol "#" is used to represent the null affix.

$\underline{\text{стол}}^{*}_{Ns}$	$\underline{\text{столы}}^{*}_{Np}$
$\underline{\text{стол}}^{*}_{As}$	$\underline{\text{столы}}^{*}_{Ap}$
$\underline{\text{стола}}^{*}_{Gs}$	$\underline{\text{столов}}^{*}_{Gp}$
$\underline{\text{столу}}^{*}_{Ds}$	$\underline{\text{столам}}^{*}_{Dp}$
$\underline{\text{столом}}^{*}_{Is}$	$\underline{\text{столами}}^{*}_{Ip}$
$\underline{\text{столе}}^{*}_{Ps}$	$\underline{\text{столах}}^{*}_{Pp}$

Paradigm of $\underline{\text{стол}}^{*}$

Fig. 3-5

"стол"	"столы"
"стола"	"столов"
"столу"	"столам"
"столом"	"столами"
"столе"	"столах"

Reduced Paradigm of $\underline{\text{стол}}^{*}$

Fig. 3-6

Given Matejka's subdivision, it is possible to reduce the multiple usages. Animate nouns ending in "#" are nominative singular and inanimate nouns ending in "a" are genitive singular. With the division, the identification for студент and стол are:

"студент" represents: $\underline{\text{студент}}^{*}_{Ns}$,

"студента" represents: $\underline{\text{студента}}^{*}_{As}$ and $\underline{\text{студента}}^{*}_{Gs}$,

"стол" represents: $\underline{\text{стол}}^{*}_{Ns}$ and $\underline{\text{стол}}^{*}_{As}$,

"стола" represents: $\underline{\text{стола}}^{*}_{Gs}$.

For each of the three morphological types, Matejka further constructed a set of tables which list the lexical attributes for every desinence in every class[†] (Fig. 3-7).

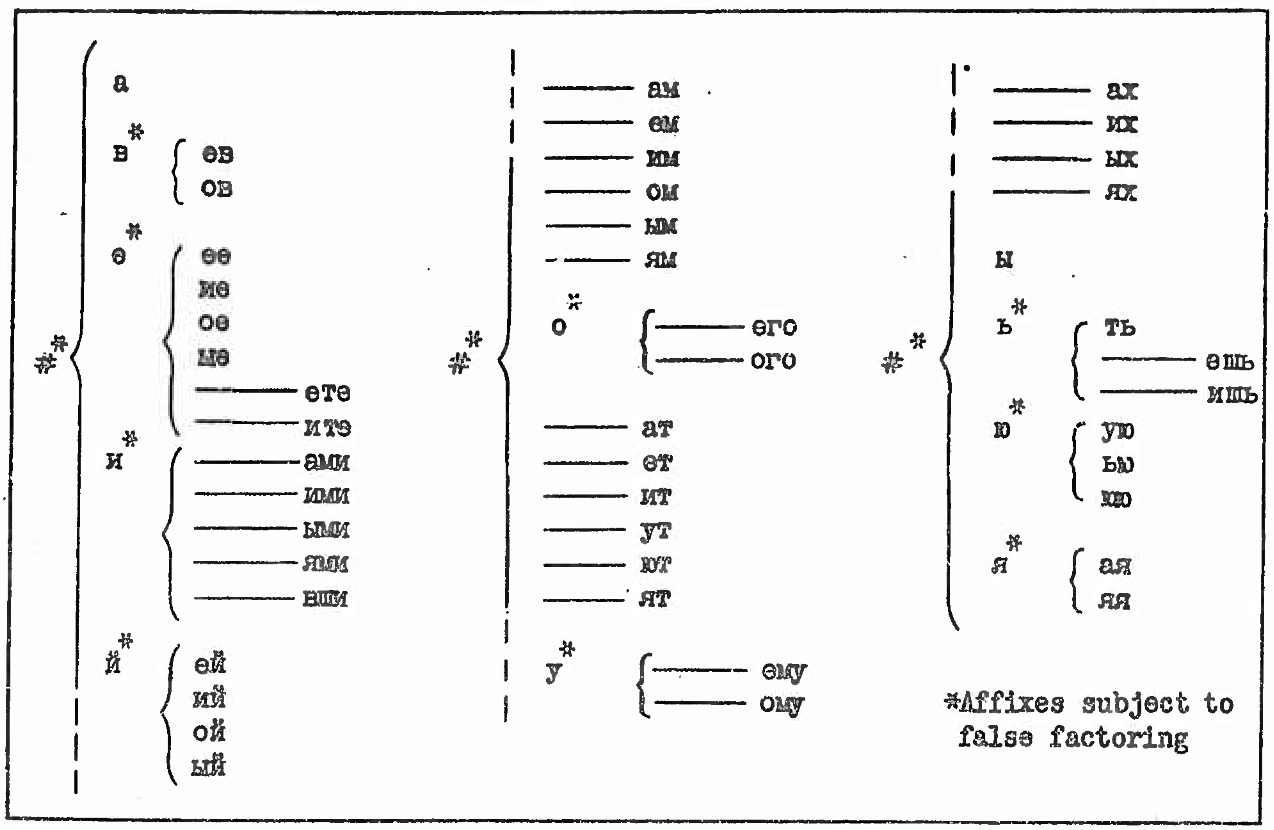
The result of these efforts was a complete definition of the paradigms of Russian inflected words, including a table of lexical attributes for the different members of the paradigm. Whereas Magassy and Matejka

[†] Errors in the set of tables are listed in Appendix B.

	N2 i1	N2 a1	N2 i2
11 и	NpAp	Np	PsNpAp
12 я	Gs	GsAs	Gs
13 ю	Ds	Ds	Ds
14 й	NsAs	Ns	NsAs
15 ев	Gp	GpAp	Gp
16 ем	Is	Is	Is
17 ях	Pp	Pp	Pp
18 ям	Dp	Dp	Dp
19 ями	Ip	Ip	Ip

Grammatical Specifications for Noun Paradigms of Class N2:
Inanimate, Type 1; Animate, Type 1; and Inanimate, Type 2 (Ref. 10)

Fig. 3-7



The Affixes of Order One Generated by the Inverse
Inflection Algorithm

Fig. 3-8

stopped at this point, the information they obtained could have been used to generate automatically a matrix $\underline{\Pi}$ (Sec. 2.3) for every word paradigm by first generating the complete paradigm instead of the reduced paradigm and then storing the lexical attributes with each member of the complete paradigm in the paradigm generating routines.[12,13] The grammatical specifications, as illustrated in Fig. 3-7, are a graphical representation of the set of vectors of lexical attributes for each desinence $\underset{\sim}{u}_m(x)$ (Sec. 2.2).

B. The Inverse Inflection Algorithm

Oettinger's <u>inverse inflection algorithm</u>[7,14] is the arbitrary factoring algorithm currently used to factor affixes for dictionary compilation and for the Continuous Dictionary Run. This algorithm provides a two-step process for factoring affixes from Russian words. As a first step, one of three affixes, "#" (null affix), "сь", and "ся", is recognized. These affixes are referred to as <u>affixes of order zero</u> and generally describe the reflexive and reciprocal properties of Russian verbal forms. As a second step are recognized fifty-seven affixes (Fig. 3-8) that are referred to as <u>affixes of order one</u>. These affixes closely coincide with the desinences of Russian words. Every Russian word has an affix of order zero and an affix of order one. If nothing is factored, then the affix "#" is assigned to the word.

The inverse inflection algorithm operates efficiently on noun and adjective paradigms, which usually require only one stem entry in the dictionary. The factorization of affixes in verb paradigms is less efficient than the factorization of affixes in noun and adjective paradigms, and generally three or four stems are required to define a paradigm completely.

To separate the grammatical functions of the stems, more extensive coding of the verb entries than of the noun and adjective entries has proved necessary (see Sec. 3). The inclusion of six more affixes would reduce the number of verb stems significantly (Table 3-1). The suggested affixes are "ьте", "йте", "л", "ла", "ло", and "ли". When only the most populous verb classes, V1, V3, and V4, were considered, a rough estimate of the effect of the inclusion of these affixes indicated that the dictionary would be reduced in size by about 5% with a potentially great simplification in the coding of the verb entries. It appears upon only superficial examination that this addition would not add much to the problem of false factoring (Sec. 3B). As an example, in the paradigm of <u>основать</u>*, which is in class V3, five stems are generated: "основа", "осн", "оску", "основал", and "оскуйт" (Fig. 3-9).

	Number of Stems			Number of Stems			Number of Stems			Number of Stems	
Class	Old	New	Class	Old	New	Class	Old	New	Class	Old	New
V1	4	2	V5.1	3	2	V8.2	4	4	V13	5	4
V2	3	2	V5.2	3	2	V9	5	4	V14	5	3
V2.01	4	2	V5.3	5	3	V9.1	4	3	V15	4	2
V3	5	3	V5.4	3	2	V10	3	2	V15.1	4	3
V4	3	2	V5.41	3	2	V10.01	3	2	V15.2	4	2
V4.01	4	2	V6	4	2	V10.1	4	3	V16	6	6
V4.02	5	3	V6.1	5	3	V10.2	3	2	V17	4	3
V4.1	4	3	V6.2	5	3	V10.3	3	2	V18	7	5
V4.11	5	3	V7	5	4	V10.4	4	3	V19	4	3
V4.2	3	2	V8	4	3	V11	5	4	V20	4	3
V4.21	4	2	V8.1	4	4	V11.1	3	2	V21	7	5
V5	4	2	V8.11	5	5	V12	5	3			

The Reduction in the Number of Verb Stems per Class if Affixes "ьте", "йте", "л", "ла", "ло", and "ли" were Included in the Inverse Inflection Algorithm

TABLE 3-1

"основа-ть"	"основал-а"
"осн-ую"	"основал-о"
"осну-ешь"	"основал-и"
"осну-ет"	"осну-й"
"осну-ем"	"оснуйт-е"
"осну-ете"	"осну-я"
"осну-ют"	"основа-в"
"основал-#"	"основа-вши"

Reduced Paradigm of <u>основать</u>*
Using the Inverse Inflection Algorithm

Fig. 3-9

"основа-ть"	"основа-ла"
"осн-ую"	"основа-ло"
"осну-ешь"	"основа-ли"
"осну-ет"	"осну-й"
"осну-ем"	"осну-йте"
"осну-ете"	"осну-я"
"осну-ют"	"основа-в"
"основа-л"	"основа-вши"

Reduced Paradigm of <u>основать</u>*
Using the Suggested Modified
Inverse Inflection Algorithm

Fig. 3-10

If the suggested affixes were factored, only three stems would remain: "основа", "осн", and "осну" (Fig. 3-10).

3. Mapping of Desinences Onto Affixes

It is convenient to determine the lexical attributes associated with each of the set of affixes for each class of words before the programs (Sec. 5) which analyze the words are considered. As in the case of the idealized dictionary (Sec. 2.2), Matejka's tables of lexical attributes are given in terms of desinences rather than of affixes, and it is necessary to map the set of desinences onto the set of affixes in order to determine the relationship between the affixes and the lexical attributes.

The procedure that is followed approximates the procedure of obtaining $\underline{V}$ from $\underline{V}^*$ in the idealized dictionary, in particular, steps 10

to 21 in Program 2-1. The technique varies from that used in Program 2-1, since the information available as input is somewhat different from the idealized case.

Several approaches other than the one to be followed could be used to obtain the "affix-lexical attribute" relationships. One that was mentioned in Sec. 2 consists of modifying the paradigm inflection routines, so that complete paradigms rather than reduced paradigms are generated. The lexical attributes are associated immediately with the generating affixes of the members of the paradigm rather than with the desinences. Although conceptually simple, this approach would involve extensive recoding of present programs.

A second possible approach is suggested by the idealized dictionary. It was pointed out in the summary of Chap. 2 that a major defect of the idealized dictionary was the amount of storage space required. The main difficulty is the fact that each desinence maps onto so many affixes that the entry function vectors, which are stored in each dictionary entry, require hundreds of bits. Since, for a given class of words, most of these bits are never used, a practical solution would be to increase the number of reference matrices, so that there is a reference matrix for each of Magassy's morphological classes. The number of columns in each matrix would be drastically reduced, since only a small number of the affixes, approximately twenty, would be used within any one class. Thus, the entry function vectors would be correspondingly reduced, and the simple identifying procedure of Program 2-3 could be used with only slight modification.† This solution

---

†This approach was suggested by D. W. Davies of the National Physical Laboratory, Teddington, England, during his visit to Cambridge, Mass. in December, 1959.

would not be practical on the Univac I, the computer currently being used at Harvard University, since this computer is not a binary machine and the individual bits are not accessible to the programmer. It would be ludicrous to simulate a binary machine by using a character position to represent a single bit.

In the scheme to be described in this section, it is a moot question whether the desinences or the generating affixes are being mapped onto the affixes. The actual process involves both, since on the one hand the generating affixes will be manipulated to determine the affixes, but on the other hand the lexical attributes which are associated with the desinences will be assigned to the generating affixes.

This section has been divided into two parts, the first dealing with the mapping technique (Sec. 3A) and the second dealing with the problems that evolved from the adopted procedure as well as with their solution (Sec. 3B).

A. Correlation of Generating Affixes and Affixes

The generating affixes are mapped onto the affixes for each of Magassy's morphological classes. Later the lexical attributes associated with the desinences will be associated with the generating affixes. This information, together with the results of the mapping operation, will determine the program for a logical tree for each morphological type. One of these trees will be scanned every time a dictionary entry is analyzed (see Sec. 5). Although the programming for a tree is more complex than that for Program 2-3, the time needed for analysis will be of the same order of magnitude, since only a minute section of the tree will be scanned during the analysis of any given word.

The technique for mapping Magassy's generating affixes onto the affixes defined by the inverse inflection algorithm is shown in Program 3-1 for a generating affix g in a class c of one of the three morphological types. This technique can be used with any system of morphological classes and any factoring algorithm. A vector $\underline{a}$ is used (not necessarily the $\underline{a}$ of Chap. 2), each of whose components is an affix factored by the inverse inflection algorithm, and which includes every affix once and only once, the order of the components being immaterial.

Symbol	Function
g	Generating affix being mapped
$\underline{\gamma}, \underline{\zeta}$	Column vectors
$(\underline{b}_c)^i$	A possible ending of the generating stem in word class c
$F(x)$	Inverse inflection algorithm on word x
$\underline{\theta}$	Affixes onto which g can be mapped

Definition of Symbols

TABLE 3-2

1 $\rightarrow \quad \underline{\gamma}^{\epsilon} \leftarrow g$

2 $\quad \underline{\zeta}^{\epsilon} \leftarrow \backslash (\underline{b}_c)^{\epsilon}, \underline{t}^{\nu(g)}, \underline{\gamma}^{\epsilon} \backslash$

3 $\quad \underline{x}_{\epsilon} \leftarrow \left[F(\underline{\zeta}^{\epsilon}) / \underline{\zeta}^{\epsilon} \right]$

4 $\quad \underline{\theta} \leftarrow \left\{ (\underline{x} \leftarrow \underline{a}) \neq 0 \right\} / \underline{a} \quad \rightarrow$

Program for the Mapping of Generating Affixes onto Affixes

Program 3-1

Every component of a column vector $\underline{\gamma}$ is defined in step 1 as the generating affix g that is being correlated. This vector is adjoined to the column vector $\underline{b}_c$ in step 2, where each component of $\underline{b}_c$ represents one of the possible endings of the generating stem from Magassy's tables in class c. (Dashes have been used in the representation of $\underline{b}_c$, since it might be necessary to know more than the last letter in each component.) The effect of this operation is to attach the generating affix to each possible generating stem.

For class N2 (Figs. 3-4 and 3-7) and generating affix "я",

$$\underline{\gamma} = \begin{bmatrix} я \\ я \\ я \\ я \end{bmatrix}, \quad \underline{b}_c = \begin{bmatrix} \text{---а} \\ \text{---е} \\ \text{---и} \\ \text{---о} \end{bmatrix}, \quad \text{and } \underline{\zeta} = \begin{bmatrix} \text{---ая} \\ \text{---ея} \\ \text{---ия} \\ \text{---оя} \end{bmatrix}$$

In step 3, every component of $\underline{\zeta}$ is considered a vector and is factored by the inverse inflection algorithm F, and the resulting logical vector is used to compress the component itself. Every compressed component is considered as a component of the row vector $\underline{x}$. The affix vector $\underline{a}$ is mapped onto $\underline{x}$ in step 4, and the resultant vector is used to reduce $\underline{a}$, giving a vector $\underline{\theta}$, each of whose components is one of the affixes that correlates with the generating affix g. In the same example,

$$F(\underline{\zeta}^{\epsilon}) = \begin{bmatrix} \text{---11} \\ \text{---01} \\ \text{---01} \\ \text{---01} \end{bmatrix}, \quad \underline{x} = \left[\text{ая, я, я, я} \right] \quad \text{and } \underline{\theta} = \left[\text{ая, я} \right].$$

The results of the mapping operation are shown in Appendix C.

B. False Factoring

The factoring algorithm, under certain conditions, can factor part of a stem with the desinence to obtain the factored affix. The residue of the word, the factored stem, will be shorter (will contain fewer characters) than the factored stem of another member of the same paradigm where this phenomenon does not take place. In such cases the canonical stem is not unique. In the example of Sec. 2.3, "атоме" will be factored into the stem "атом" and the affix "е" while "атому" will be factored into the stem "ат" and the affix "ому". Likewise, if the factoring algorithm cannot factor the entire desinence, a factored stem can be longer than the normal factored stem of a paradigm. For example, while "оснуй" is factored into "осну" and "й", "оснуйте" is factored into "оснуйт" and "е" (Fig. 3-9). Both extra long and extra short stems, which will be referred to as anomalous stems, exist in the Harvard Automatic Dictionary.

Anomalous stems are a natural consequence of factoring even in the idealized dictionary, since independent of coded syntactical information, it is impossible to write a factoring algorithm that will recognize whether or not a string of letters represents some desinence. In the case of the idealized stem dictionary, an extra dictionary entry is generated with its own entry function vector, whenever an anomalous stem occurs. Similarly, in the Harvard Automatic Dictionary each anomalous stem generates its own dictionary entry. The difficulty lies in the fact that, prior to this work, there was no information in the experimental dictionary equivalent to the entry function vector to indicate that a stem is anomalous. This lack of information was the cause for an excessive number of stem homographs. Since

many of these homographs from the experimental dictionary do not appear as homographs in the idealized dictionary, they are nonessential homographs in the experimental dictionary.

An example of the nonessential stem homography in the experimental dictionary is shown by the reduced paradigms of two Russian nouns, <u>вал</u>* and <u>валюта</u>* (Figs. 3-11 and 3-12). The string "валют" from the paradigm of <u>валюта</u>* is factored into the stem "вал" and the affix "ют" by the inverse inflection algorithm. This stem is identical with the stem of <u>вал</u>*. Therefore, any time that either any member of the reduced paradigm of <u>вал</u>* or the paradigmatic form "валют" appears in a text, both dictionary entries with the stem "вал" are selected. In the idealized dictionary different affixes would be represented in the two entry function vectors, so that one of the dictionary entries would always be incompatible.

Another problem occurs in the paradigm of <u>атом</u>* (Fig. 3-13). Two distinct stems are factored in this paradigm, and the affix "ом" can be associated with both of them. The affix "ом" is factored both from the string "атом" and from the string "атомом". It is therefore necessary to

"вал-#"	"вал-ы"
"вал-а"	"вал-ов"
"вал-у"	"вал-ам"
"вал-ом"	"вал-ами"
"вал-е"	"вал-ах"

Reduced Paradigm of <u>вал</u>*
Fig. 3-11

"валют-а"	"валют-ей"
"валют-ы"	"вал-ют"
"валют-е"	"валют-ам"
"валют-у"	"валют-ами"
"валют-ой"	"валют-ах"

Reduced Paradigm of <u>валюта</u>*
Fig. 3-12

"ат-ом"	"атом-ы"
"атом-а"	"атом-ов"
"ат-ому"	"атом-ам"
"атом-ом"	"атом-ами"
"атом-е"	"атом-ах"

Reduced Paradigm of атом*

Fig. 3-13

be able to determine when the affix "ом" should be the resultant affix of the desinence "ом" and when it should be the resultant affix of the desinence "#". This is an example of the artificial affix homograph. This type of homograph is also nonessential, since in the idealized dictionary the appropriate lexical attribute would be listed with the affix in both cases.

As is shown in Appendix C, every affix appearing in the fourth column is the result of a factoring that produced an anomalous stem; for instance, in the paradigm of валюта* (class N4) only the form "валют" is factored into an anomalous stem.

The following is a discussion of the various operations that have been adopted to patch the experimental dictionary so that its output should be identical with the output of the idealized dictionary.

When a single affix is associated with an anomalous stem, the entry function vector $\underline{y}_k$ for the anomalous stem contains only one "1". It is a simple matter to put a mark somewhere in the existing dictionary entry to indicate that the item should be treated as a fully inflected item. Then the affix of the text word whose stem matches the stem of the dictionary entry should be compared with the single affix stored in the dictionary.

Giuliano[1] already adopted such a technique with respect to stems with zero or one letter to reduce homography. If the experimental dictionary affix is not identical with the affix of the text word when the special mark is present, then the dictionary entry is incompatible, that is, it is not the one being sought.

It should be pointed out that during dictionary compilation in the experimental system, when the inflected forms are generated from the canonical forms, they are generated in the order given in the reproduction of Magassy's table in Ref. 7, as illustrated in Fig. 3-4. When these paradigms are condensed by a later routine, the affix from the first form encountered with a given stem is stored in the dictionary. It is indeed fortunate that the affix normally stored with the generating stem never causes confusion with the affixes that form anomalous stems. In particular, it is fortunate that the form "атомом" is not the first form with the stem "атом" that is generated, since if it were, the affix "ом" would be stored in the dictionary entry of "атом", while "ом" is already stored with the dictionary entry of "ат", originating from the form "атом". If "атомом" were the first generated form with the stem "атом", and "атом" were the first generated form with the stem "ат", then there would be no automatic way of distinguishing that the latter stem is the anomalous one.

There remains a small group of noun paradigms, such as that of <u>атом</u>*, which requires special treatment because there is more than one affix associated with an anomalous stem; for example, both "ом" and "ому" are factored, leaving the stem "ат". Since there is no coding present in the experimental dictionary entry to distinguish the different inflected forms, and since fortunately there appear to be never more than two affixes associated with

an anomalous stem, an extra dictionary entry can be generated, and each of the two anomalous stem inflected forms can be treated as a fully inflected item, thereby increasing the size of the dictionary by only 0.5%. This increase would not occur in the idealized dictionary, because the entry of the stem "ar" would contain all the necessary information about both affixes.

Among the verb paradigms in the experimental dictionary there are many more anomalous stems, owing to a large number of verb desinences that are not factored by the inverse inflection algorithm. If the paradigm of <u>подходить</u>[*] is used as an example, four unique stems are generated: "подход", "подходи", "подходил", and "подхож". These stems have seven, three, four, and one affixes associated with them, respectively (Fig. 3-14). Only in the stem "подхож" did it seem practical to mark the stem as the noun and adjective anomalous stems were marked, since so many affixes are associated with the other stems. If the same system were adopted with the other stems, the

"подходи-ть"	(B0)	"подходил-#"	(B3)
"подхож-у"	(B1)	"подходил-а"	(B3)
"подход-ишь"	(B1)	"подходил-о"	(B3)
"подход-ит"	(B1)	"подходил-и"	(B3)
"подход-им"	(B1)	"подход-и"	(B4)
"подход-ите"	(B1)	"подход-я"	(B5)
"подход-ят"	(B1)	"подходи-в"	(B6)
		"подходи-вши"	(B6)

Reduced Paradigm of <u>подходить</u>[*]
with Associated Tense and Mood Indicators

Fig. 3-14

size of the dictionary would increase by an intolerable amount, thus defeating the main advantage of a stem dictionary over a full paradigm dictionary.

The problem has not become acute since, when the dictionary was being compiled, it was recognized that the multiplicity of stems occurring in every verb paradigm would cause stem homographs. A coding scheme, the tense and mood indicators, was incorporated into the dictionary entries to identify the grammatical functions that the stem and any of its affixes could assume (Table 3-3). The correct coding associated with the inflected forms in Fig. 3-14 has been placed in parentheses next to each inflected form. The coding, as it would appear in the third semiorganized word for each stem, is shown in Fig. 3-15.

B0 – infinitive
B1 – present indicative
B2 – future indicative
B3 – past indicative
B4 – imperative
B5 – past gerund
B6 – present gerund

Tense and Mood Indicators in the Third Semiorganized Word of Verb Entries

TABLE 3-3

"подход"	B1B4B5
"подходи"	B0B6
"подходил"	B3
"подхож"	B1

Tense and Mood Coding in the Third Semiorganized Word for the Stems of подходить

Fig. 3-15

As an example of how the tense and mood coding helps in analysis, consider the reduced paradigm of the noun <u>подход</u>* (Fig. 3-16). Stem

"подход-#"	"подход-ы"
"подход-а"	"подход-ов"
"подход-у"	"подход-ам"
"подход-ом"	"подход-ами"
"подход-е"	"подход-ах"

Reduced Paradigm of <u>подход</u>*
Fig. 3-16

homography exists with the stem "подход", which is common to both the noun and verb paradigms. There is no essential homography in the experimental dictionary, since all the affixes associated with the two stems "подход" in the two paradigms are different. For example, if the string "подхода" occurs as a text word, both stems "подход" will be selected during dictionary look-up. The affix "a" in class V4 represents the single grammatical function past indicative, but the past indicative cannot be associated with the verb stem "подход", as shown by the fact that there is no "B3" coded in its third semiorganized word. Therefore "подход" cannot be an inflected form of the verb paradigm. However, since the string contains an affix that can belong to the paradigm of the noun <u>подход</u>*, the string can be correctly identified as a noun.

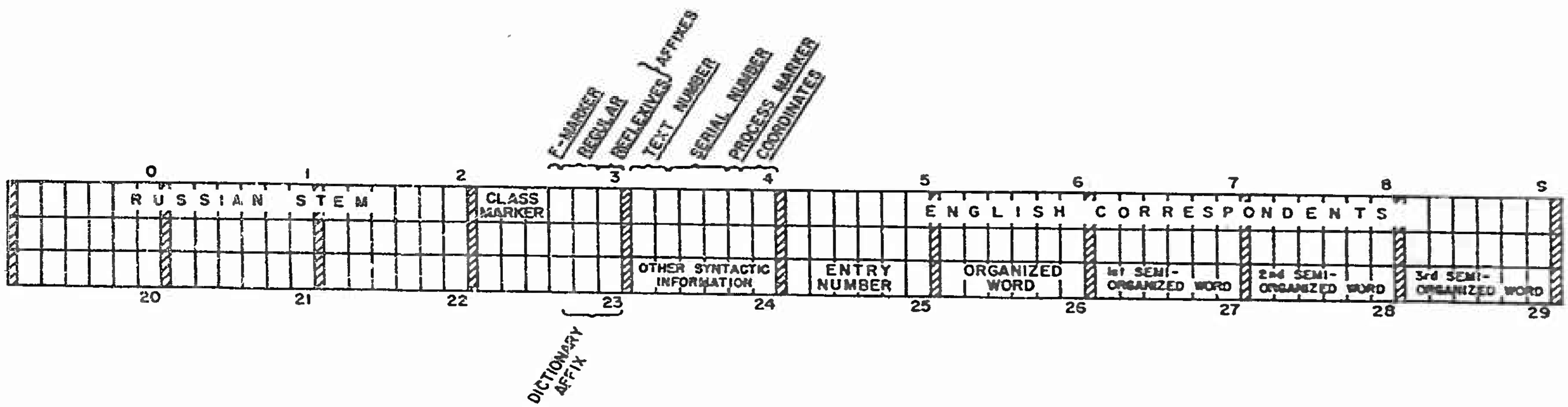
In the idealized dictionary the special coding would not be necessary, since the lexical attributes would be represented in the reference matrix and the entry function vector.

4. The Anomalous Stem Program

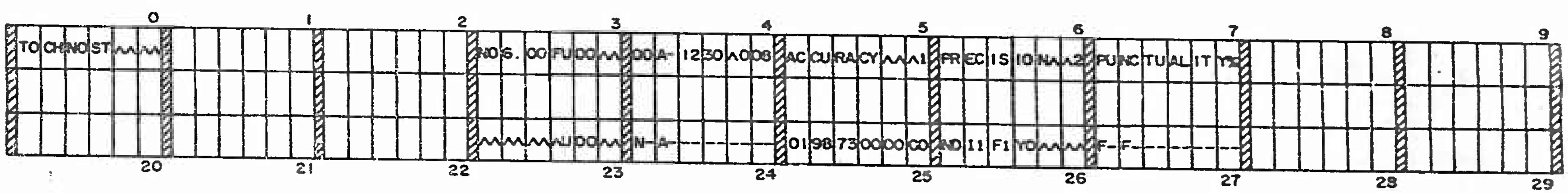
Anomalous stems are detected and marked automatically in the experimental dictionary, so that the analyzing programs (see Sec. 5) can recognize that only the single affix that is stored in the dictionary entry is associated with the stem. This serves two distinct purposes: (1) The affix stored in a marked dictionary entry is compared with the affix of the text word. If they do not match, the dictionary item is incompatible. (2) The mark indicates that the affix stored in the dictionary was caused by false factoring.

The anomalous stem program has three outputs: (1) a tape containing all the input items with an appropriate mark in the anomalous stem items, (2) a list of potential dictionary entries generated by the program (Sec. 4A), and (3) a list of potential dictionary entries which must be studied further (Sec. 4B).

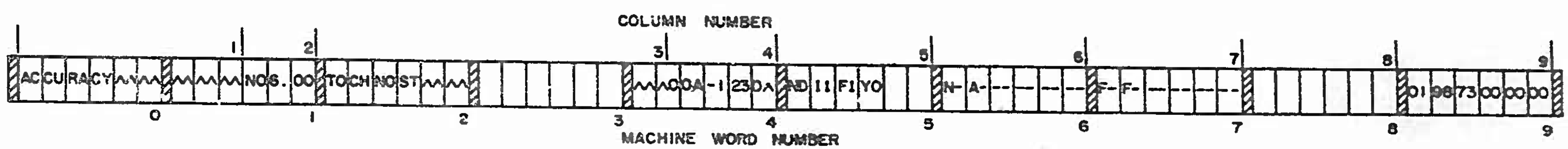
Every dictionary entry is stored as a 30-word item,[1] a size chosen both to be compatible with the block transfer operations (60 words at a time) of the Univac I and to have sufficient space available to store the necessary syntactic and lexical information and various forms of experimental markings. Since the morphological and syntactic information is contained in fewer than ten of these machine words, it has been feasible to compress the information of immediate interest into 10-word items, which will be referred to as texthadic items.[15,16] An analyzed 30-word item and the condensed texthadic item are illustrated in Fig. 3-17. The columnar layout of the texthadic, with reference to the 30-word item, is listed in Table 3-4. The anomalous stem program will be described in terms of the 30-word item.



Format of 30-word Item



A Sample 30-word Item



A Sample 10-word Item

Format of a 30-word Item and a 10-word Item

Fig. 3-17

Column 1 - First English equivalent from dictionary. (Word 5)
Column 2 - Class marker. (From Word 3)
Column 3 - Russian word transliterated with affix attached. (Words 0-2)
Column 4 - Text serial number. (From Word 4)
Column 5 - Organized word. (Word 26)
Column 6 - First word of interpreted information. (Word 24)
Column 7 - Second word of interpreted information. (Word 27)
Column 8 - Third semiorganized word. (Word 29)
Column 9 - Dictionary serial number. (Word 25)

Columnar Layout of Texthadic with References to 30-word Item

TABLE 3-4

Only the first English correspondent from the 30-word dictionary item is transferred to the 10-word texthadic item. This correspondent has little significance in the translations of the examples that will be given throughout this thesis. The purpose of including the single correspondent in the texthadic items is to help the reader who has no knowledge of Russian to identify individual Russian words.

The program has been designed with two purposes in mind, first, to update the entire experimental dictionary when a change is made in the word analyzer program and, second, to process new items before they are merged into the existing experimental dictionary. It should be stressed once more that this program would not be necessary in the idealized dictionary, since the necessary markings exist in the reference matrix and the entry function vector.

A. The Identification of Anomalous Stems

Russian words are considered to be divided into six morphological types (Table 3-5), with a distinct format for the representation of the grammatical properties of each type. These six types are represented by appropriate alphabetic symbols in character position 1 of word 3 of the 30-word item as shown in Table 3-5.

(1)	Noun (N)
(2)	Adjective (A)
(3)	Verb (V)
(4)	Pronoun (P)
(5)	Numeral (D)
(6)	Indeclinable (I)

Morphological Types in the Harvard Automatic Dictionary
TABLE 3-5

The program examines the class marker and affix of every 30-word item. The logic of the program is expressed in a tree. The program branches initially on the three productive morphological types, the noun, adjective, and verb. The secondary branch for each type is on the various classes among which anomalous stems can occur. The third and last branch is on the affixes that are factored with anomalous stems. One of these affixes is stored in the dictionary during compilation. The classes among which anomalous stems occur can be identified easily, since they are the classes with affixes in the fourth column of the table of Appendix C. A complete list of these affixes from Appendix C is shown in Table 3-6.

Class	Affixes	Class	Affixes
A3	ам ат в ев ем ет им ит ов ом ут ым ют ям ят	N8	ам ат ах в ев его ем[1] ему[1] ет[2] ете[2] им ит[3] ите[3] их ов ого ом[4] ому[4] ут ым ых ют ям ят ях
A4	вши	N8.1	ом
A5	е й	N8.15	ем им
A8	ах вши их ых ях	N10	ие[1] ий[1]
N (any)	сь ся	N11	ей ью
N1	ам ат в ев ем[1] ему[1] ет[2] ете[2] им ит[3] ите[3] ов ом[4] ому[4] ут ым ют ям ят	N11.1	ий ью
N1.1	ах их ых ях	N11.2	ью
N1.2	ев ем ет ов ом	V (any)	е
N2	ая ее[1] ей[1] ие[2] ий[2] ое[3] ой[3]	V1	ая яя
N2.1	ей ью	V3	ую
N3	ете ите ть	V4	ами ая ими у ую ыми ями
N3.05	ть	V4.01	у
N4	ам ат в ев ем[1] ему[1] ет[2] ете[2] им ит[3] ите[3] ов ом[4] ому[4] ут ым ют ям ят	V4.02	ая ей ой ую
N4.05	ем	V4.1	у
N4.06	ом	V4.11	у
N4.1	вши ах их ых ях	V5.2	ами ими ыми ями
N5	ами ете ими ите ть ыми ями	V5.3	ая ей ий ой ую ый яя
N5.05	ая ее[1] ей[1] ое[2] ой[2] ую ые[3] ый[3] яя	V6.1	ой
N5.2	ей[1] ью[1]	V6.2	ами ими ыми ями
N5.3	ий ью	V10.1	ими
N6	ть	V10.4	ими
N6.1	ешь ишь	V12	ая
N7	ий	V13	ей ью
		V14	ой
		V15	ую
		V18	ам

Affixes Marked by Anomalous Stem Program
(Superscripts denote automatically generated pairs.)

TABLE 3-6

A 30-word item containing an anomalous stem is marked with a "1" in character position 12 of the organized word, word 26 of the 30-word item. If the program finds that the 30-word item has two affixes associated with the stem, as in "ат-ом", it automatically generates the second member of the pair, in this case "ат-ому", on a separate tape. A "1" is inserted into character position 12 of the organized word, an "M" is inserted into character position 4 of word 4 to identify the source of such an entry, and if there is an "F" in character position 7 of word 3, indicating that this is a canonical form, that is, the form from which the word was generated, the "F" is deleted. This output then can be inserted with other new entries into the dictionary in a single pass.

To facilitate changes in the program, any previous "1" in character position 12 of word 26 that was inserted by previous versions of this routine is erased. This makes it possible to update the dictionary quickly if the program has to be altered.

When the Harvard Automatic Dictionary was first modified by this program in November 1959, only 89 blocks due to anomalous stems with two affixes had to be added to the 15,000 blocks which existed at the time.

B. Exceptions

Among the verb paradigms that have been assigned to classes V4, V4.01, V6.2, and V8, there exist several where the desinences "ь" or "я" are factored together with part of a stem ending in "с", as with "све-сь", an imperative form of the verb свесить*. The inverse inflection algorithm factors an affix of order zero and then possibly an affix of order one. In the above example the stem is "св", the affix of order one is "е", and the

affix of order zero is "сь". These verbal forms must be identified, so that they will not be analyzed as ordinary reflexive forms.

Because of the large number of reflexive verbs, it is too much of a burden for the anomalous stem program to identify automatically these rare nonreflexive inflected forms. A special policing subroutine of the anomalous stem program prints out on the third output tape of the program all the stems in these classes that end in "с". These stems then can be inspected visually, and a "2" can be inserted into character position 12 of the organized word of those 30-word items which contain such a special anomalous stem. For example, if the paradigm of <u>свесить</u>* (Fig. 3-18) is considered, only the stem "свес" can be identified automatically. Once the stem "свес" is found, the entire paradigm can be studied, and the appropriate anomalous stem "св" marked.

One other potential problem is treated by this policing subroutine. The generating stems of the verbs in class V7 were not defined in sufficient

"свеси-ть"	"свесил-а"
"свеш-у"	"свесил-о"
"свес-ишь"	"свесил-и"
"свес-ит"	"св-е-сь"
"свес-им"	"св-е-ся"
"свес-ите"	"свеси-в"
"свес-ят"	"свеси-вши"
"свесил-#"	"свесьт-е"

Reduced Paradigm of <u>свесить</u>*
Fig. 3-18

detail by Magassy to determine the affixes that might be factored from the form containing the null generating affix (Fig. 3-19). The canonical form

CLASS	EXAMPLES	CLASS IDENTIFICATION
V7	мерзнуть возникнуть	A class embracing the verbs: 1. ending in нуть; 2. losing ну in the masculine past tense forms.

GENERATION RULES	
Generating Stem (GS)	Generated Forms with Specified Generating Affixes
word - last four letters.	a. word b. GS + ну c. + нешь d. + нет e. + нем f. + нете g. + нут h. + # i. + ла j. GS + ло k. + ли l. + ни m. + ните n. + нь o. + ньте p. + нув q. + нувши r. + ши

Definition and Description of Class V7

Fig. 3-19

of every new verb in class V7 is also printed out, making it possible to identify a verb where an affix other than the null affix would be factored from a stem with a null desinence. Such a form, which is an anomalous stem, is also marked with a "2" in character position 12 of the organized word.

When the dictionary of approximately 30,000 entries was initially scanned with this policing program, only seven entries had to be marked with a "2" in character position 12 of word 26. None of the seven entries was in class V7.

5. The Word Analyzer Programs

Three machine programs have been written to interpret the affixes of nouns, adjectives, and verbs (the noun analyzer, adjective analyzer, and verb analyzer, respectively). The nonproductive classes of words, those in which there is a limited and known number of words, such as pronouns and numerals, are processed by the adjective analyzer. Three separate programs were written only because of the restrictive size of the internal memory of the Univac I. Conceptually, the three programs are a single comprehensive program.

Since look-up requires a distinct run preceding the three affix interpreting programs, it is necessary under present conditions to copy every item found in the dictionary onto an output tape, even though it is known that about 20% of the items will be eventually rejected during the homograph deletion phase (Fig. 3-1). These extra items have to be processed several times before they are eliminated: in dictionary look-up, in the three analyzer passes, in sorting back to text order, and finally in deleting homographs. By far the most time-consuming of these passes is the sorting run.

If a larger internal memory were available, the present decomposition into many separate programs would not be necessary. The 30-word items could be analyzed at the same time as they were being looked up in the dictionary. In the event that a homographic set were looked up, only the correct member or members of the set would be kept. If all the members of the set were incompatible, an artificial 30-word item could immediately be generated to indicate that fact. Conceptually, therefore, dictionary look-up, the

analyzing runs, and the homograph delete run could be carried out in a single pass, and indeed this is possible on any of the several machines now available with a larger memory than that of the Univac I.

The three affix analyzer programs have been made as uniform as possible. The same symbols have been used in the three flow charts to describe the actions of each program (Appendix D). The grammatical functions, as determined on a word-by-word basis, are stored in words 24 and 27 of the augmented texts (Fig. 3-17). The arrangement of grammatical information for nouns, adjectives, and verbs will be given in Tables 3-7 to 3-9. The format for pronouns has been described by Matejka[17] and Coppinger,[18] and the format for numerals by Magassy.[19] Matejka[20] has illustrated the format for prepositions, one of the classes of indeclinable words.

If a 30-word item is found to be incompatible, that is, the stem and affix of the text word do not correspond to the dictionary entry that was found by stem comparison, this is indicated by the same set of marks by all three analyzer programs. In terms of the idealized stem dictionary, an incompatible item would be one whose reduced lexical attribute vector is all zeros. An example of an incompatible item was shown in the last example of Sec. 3 of this chapter: "подхода" is identified by the mood and tense coding as not being an inflected form of a verb paradigm. To eliminate such a 30-word item from further consideration in syntactic analysis, the symbol "INCOMPAT A" is put into word 24.

Several other similar symbols are used. Since an indeclinable word has a one member paradigm, an indeclinable item is incompatible if the affix stored in the dictionary is not identical with the affix of the text word. The symbol "INCOMPAT I" is used to denote this condition. Adjectives and

verbs are tested for voice (affixes of order zero). If the voice coding is inconsistent with the affix of order zero, then the symbol "INCOMPAT R" is placed in word 24. Lastly, the symbol "INCOMPAT Z" is placed into word 24 if the item belongs to a class that cannot be analyzed automatically and is indicated by a class marker greater than 75.

It should be noted that "INCOMPAT A" is of a higher priority than "INCOMPAT R"; that is, if an item is incompatible in the sense of both the symbols "INCOMPAT A" and "INCOMPAT R", the former symbol is placed in word 24. An item with a class marker greater than 75 can be marked "INCOMPAT A" instead of "INCOMPAT Z" only if the affix of the word does not correspond to any of the affixes tested for in the various analyzer programs. (See the descriptions of the individual programs.) This priority exists because the affixes are checked first by the analyzer programs.

Since the three analyzer programs exist at present as separate programs in the Continuous Dictionary Run, they will be discussed individually.

A. Noun Analyzer Program

The noun analyzer program analyzes only noun morphological types, whose formal definition is given by the letter "N" in character position 1 of word 3 of a 30-word item. All other items on an input tape are copied directly without modification.

The logic of the program is expressed in a tree structure (Flow Chart 1 of Appendix D). The first branching within the tree is determined by the affix of the noun. The fastest way of recognizing the affix is to compare the affix of the text word with a complete list of affixes that can occur

legitimately in the various noun classes. To reduce the time spent in this search, the list has been ordered so that the most frequently occurring affixes appear at the head of the list (see Sec. 8).

After the program branches on the affix, an appropriate subtree is entered. The usual order of branching in the subtree, as a matter of efficiency, is by class marker and then by character positions 3 and 4 of the organized word. To reduce the number of instructions in the program, the integral component of the class marker is identified before the fractional component. Similarly, the fourth character position of the organized word is usually tested prior to the third character position.

Character position 3 describes if the noun is animate or inanimate. Character position 4 divides the animate or inanimate classes further. By this subdivision the 38 classes created by the class markers are increased to 108, that is, there are 108 distinct paradigm classes for noun types. If the idealized dictionary were being used, there would be 108 different definitions of $\underline{\Pi}$ for the morphological type of nouns alone.

Before the analysis of the noun is started, the word is tested for an anomalous stem, which is signified by a "1" in character position 12 of the organized word. If a "1" is found, then the affix of the text word is compared with the affix stored in the dictionary entry. If there is no match, this means that the dictionary item cannot represent the text word. The item is labeled "INCOMPAT A" and the process is terminated. If there is a match, or if the word is not an anomalous stem, the analysis of the word is started. Throughout the tree there are further tests for anomalous stems at various levels of branching.

If a terminal of the tree, indicating compatibility between the stem and affix, is reached, the case and number is entered in word 24 of the 30-word item, where a character position is reserved for each case and number combination (Table 3-7) (also see Fig. 3-17). The case coding was chosen to be mnemonic and the machine word is divided into two sections to express number, the first six characters representing the singular and the last six the plural. The gender is inserted into word 27 in the character position corresponding to the related information on case and number (Table 3-8).

Character 1:	N	- if nominative singular
Character 2:	G	- if genitive singular
Character 3:	A	- if accusative singular
Character 4:	C	- if dative singular
Character 5:	I	- if instrumental singular
Character 6:	P	- if prepositional singular
Character 7:	N	- if nominative plural
Character 8:	G	- if genitive plural
Character 9:	A	- if accusative plural
Character 10:	C	- if dative plural
Character 11:	I	- if instrumental plural
Character 12:	P	- if prepositional plural

Format of Word 24 of Augmented Text with Information on Case and Number for Noun and Adjective Morphological Types

TABLE 3-7

M	- masculine
F	- feminine
N	- neuter
B	- masculine or neuter (adjective types only)
A	- masculine, feminine, or neuter

Allowable Characters in Word 27 of Augmented Text for Gender of Noun and Adjective Morphological Types

TABLE 3-8

The unused characters of words 24 and 27 are filled with zeros. These unused characters are sometimes changed to spaces or dashes before appearing on output lists designed for detailed linguistic study. Multiple lexical attributes are indicated by the presence of more than one identifying character in words 24 and 27.

A "C" rather than a "D" was used to represent the dative case because the letter "C", like the letters "N", "G", "A", "I", and "P", can be used as an extractor in the Univac I, but "D" cannot.

B. Adjective Analyzer Program

In addition to analyzing the adjective morphological types (participles are listed as adjectives in the experimental dictionary), which are identified by the letter "A" in character position 1 of word 3 of a 30-word item, the adjective analyzer program processes all the nonproductive morphological types of Russian words, for example, pronouns, numerals, and prepositions. The other items on an input tape are copied directly, without modification.

The logic of this program is expressed in a tree structure (Flow Chart 2 of Appendix D) similar to the tree of the noun program. After the initial anomalous stem test the first branching within the tree is determined by the affix of the adjective, and the adjectival affix list is ordered on frequency of occurrence.

After the program branches on the affix, there is only a single comparison on the integral component of the class marker in the subtree. With the exception of the anomalous stem tests, which are scattered throughout the program, this comparison determines the grammatical information

completely. When a compatible terminal of the tree is reached, the program inserts the case and number information into word 24 of the 30-word item, and the gender into word 27 of the same item in a format identical to the noun format (Tables 3-7 and 3-8).

Another set of marks is added to the 30-word items of short form and comparative adjectives. Any adjective with the affix "ee" is marked with a "1", and any adjective with the affix "e" is marked with a "2" in character position 8 of the organized word. This indicates that the adjective may function as a comparative adverb in a sentence. All short forms are marked in character position 9 of the organized word. Those with affixes "#", "a", or "ы" are marked with a "1" to indicate that the adjectives may function as verbs. Short forms with affixes "e" or "o" are marked with a "2" to indicate that the adjectives may function as verbs or adverbs. Forms with the affix "и" are marked with a "3" to indicate that they may function as adverbs. The markings are summarized in Table 3-9. The main advantage derived from this marking is that the dictionary need not be cluttered with a large number of adverbs, genuinely homographic with adjective entries.

If an adjective ending in "ee" can be used only comparatively, the mark "INCOMPAT EE" is placed in word 24 to distinguish it from other

Character 8:	1 – if adjective ends in "ee"
	2 – if adjective ends in "e"
Character 9:	1 – if adjective ends in "#", "a", or "ы"
	2 – if adjective ends in "e" or "o"
	3 – if adjective ends in "и"

Allowable Characters in Character Positions 8 and 9 of the Organized Word for Adjectival Morphological Types

TABLE 3-9

incompatible adjectival forms. Such forms are incompatible in the sense that no case, number, and gender can be assigned to them.

Only affixes of order one are compared in the adjectival tree, since the affixes of order zero refer only to the voice of the adjectives, which is tested only if the initial analysis is successful. The adjectival entries in the dictionary are marked to indicate whether the 30-word dictionary items are reflexive (R), nonreflexive (0), or both reflexive and nonreflexive items (Δ). If the symbol "R" or "0" is found (in character position 7 of word 26), the affix of order zero is checked for correspondence. If the affix matches, an "R" or a "0" is placed in character position 11 of word 26. If the affix does not match, the previous grammatical information is erased and the symbol "INCOMPAT R" is put into word 24 instead. If a reflexive affix of order zero is found, an additional test is made. Passive participles and nonparticipial adjectives cannot be reflexive, therefore character position 10 of the organized word is tested for an active participle. If an active participle is not found, the item is incompatible.

It is important to distinguish between the functions of the characters in positions 7 and 11 in the organized word. The character in position 7 indicates whether a reflexive or nonreflexive adjective is permitted by that dictionary entry, while the character in position 11 indicates whether the adjective is reflexive or nonreflexive. As an illustration, consider the typical adjective with a null affix of order zero and a delta (Δ) in character position 7 of the organized word. Sensing the delta, the program does not check whether or not the voice of the adjective is compatible. However, a zero (0) is placed into character position 11, so that a future program can immediately sense the voice of the adjective.

The last test of adjectival morphological types determines whether a word such as портной* functions only as a noun. In certain cases, depending on the affix and the animateness of the adjective (Table 3-10), the word cannot be in the accusative case. If the morphological adjective functions only as a noun, the accusative case lexical attribute can be eliminated 45% of the time.

Animate			Inanimate		
Affix	Frequency	Case and Number Eliminated	Affix	Frequency	Case and Number Eliminated
-ой	8.1%	As	-ых	9.1%	Ap
-ые	5.0	Ap	-ого	6.1	Ap
-ое	3.9	As	-их	4.6	Ap
-ие	2.8	Ap	-его	1.3	As
-ый	2.0	As		21.1 %	
-ий	1.6	As			
-ее	1.3	As			
	24.7%			Total 45.8%	

Expected Frequency of Occurrence of Affixes Which Can Reduce Ambiguity with Adjectives Used as Nouns

TABLE 3-10

Since the pronouns and numerals are nonproductive types, that is to say, there is a finite and small group of each in the Russian language, it is not practical to write a program to analyze the words. It is simpler to code the grammatical functions of these words directly when preparing the 30-word items for the dictionary.[18,19] These words are therefore stored and looked up as inflected forms. The adjective analyzer simply transposes the stored information into words 24 and 27 of the augmented text.

During look-up, indeclinable words, that is, words with the letter "I" in character position 1 of word 3, are selected on the basis of stem

comparison only. The adjective analyzer therefore compares the affixes and passes only those items where the dictionary affix and text affix match. Otherwise the symbol "INCOMPAT I" is inserted into word 24.

In addition, if an indeclinable noun or an adjectival or nominal abbreviation is found, word 24 is filled with "NGACIPNGACIP" and word 27 with "AAAAAAAAAAAA", indicating that the item might be used in any case, number, and gender whatsoever.

C. Verb Analyzer Program

The verb analyzer program, the last of the three analyzer programs, analyzes only verb items, whose formal definition is given by the letter "V" in character position 1 of word 3 of a 30-word item. All other items on an input tape are copied directly, without modifications.

The logic of this program is expressed in a tree structure (Flow Chart 3 of Appendix D) similar to the tree structures of the noun and adjective analyzer programs. After the initial anomalous stem test, the first branching is determined by the affix of the verb, which is compared with the ordered list of affixes in the program. As with adjectives, only the affixes of order one are compared. For programming ease, the subtree entered after the first branching compares first on the integral portion of the class marker and then on the fractional portion of the class marker.

If a verb is identified as being in either the present or the future indicative, the ambiguity is resolved by checking character position 2 of the organized word (Table 3-11).

In most branches of the logical tree of the verb program the lexical attributes can be determined from the affix and class marker alone. The

N	imperfective aspect
S	perfective aspect
U	momentary action (perfective)
M	iterative action (imperfective)
K	perfective or imperfective aspect

Notation of Character Position 2 of Word 26 for Verb Entries

TABLE 3-11

tense and mood coding in the third semiorganized word is used as a check to ensure that the function to be assigned to the stem is an allowable function. In a few branches, however, the tense and mood code must be used to help determine the grammatical functions. (See in particular the subtree of the affix "и".)

The markings for verbs differ significantly from those for nouns and adjectives (Table 3-12). The first six character positions in word 24 are reserved for person and number. The person and number of verbs in the present or future tenses are indicated by the appropriate character in any one of the first six initial character positions. Since for verbs in the past tense the person cannot be determined from the morphological characteristics, either all of the first three or all of the second three character positions are filled to designate number. For all verbs, the tense is given in character position 7, the gender is given in character position 8, and the mood in character position 9. The affix of the verb of order zero is checked to determine its voice, which is noted in character position 10. The only type of essential homography present within verb forms is the dual interpretation of second person plural indicative and plural imperative of some verbs ending in the string "ите". The former interpretation is displayed in word 24, but an "X" is inserted into character position 11 to denote the homography.

Characters 1-6

Option A: (Present and future tenses)

V in character position 1 = 1st person singular
Z " " " 2 = 2nd person singular
T " " " 3 = 3rd person singular
V " " " 4 = 1st person plural
Z " " " 5 = 2nd person plural
T " " " 6 = 3rd person plural

Option B: (Past tense)

SSS in character positions 1-3 = 1st, 2nd, or 3rd person singular
PPP " " " 4-6 = 1st, 2nd, or 3rd person plural

Characters 7-12

7: A = past (tense)
B = present
C = future
X = present or future

8: M = masculine (gender)
F = feminine
N = neuter
A = any

9: D = indicative (mood)
E = imperative
F = infinitive
G = gerund

10: R = reflexive (voice)
O = nonreflexive

(NOTE: This voice coding should not be confused with the same symbols used in the organized word where information is stored in advance of which voice the verb can take. This coding states the voice of the verb in each specific occurrence.)

11: X = special situation among some verbs with affix "ите" which can be both 2nd person plural indicative and plural imperative.

12: Not used

(NOTE: If a character position is not applicable, it is filled with a space. If a character position is used in the negative sense (e.g., not 1st person singular), it is filled with a zero which is later modified to a dash.)

Format of Word 24 of Augmented Text with Information on Person, Number, Tense, Gender, Mood, and Voice for Verb Morphological Types

TABLE 3-12

If a verb passes through a compatible terminal of the tree, the voice is checked for compatibility. (See Sec. 5B for the details.) The "R-O-Δ" marks are in character position 3 of word 26 of verbal forms.

It is necessary to note that character position 3 in word 26 and character position 10 in word 24 of verbal forms correspond to character positions 7 and 11 in the organized word of adjectival forms. The reason for using different character positions is purely historical. At the time that this information was inserted into the experimental dictionary, some of the character positions had already been coded with other information. These codes had to remain frozen to avoid considerable reprogramming. It would be highly desirable to use the same character positions for both adjectival and verbal forms when reprogramming the dictionary for production purposes.

The small set of verbal forms in which there is artificial factoring that generates a spurious affix of order zero (see Sec. 4B) have to be handled in a special way by the verb analyzer program. Before the main tree is entered, character position 12 of word 26 is checked for a "2". If it is found, and the text word has a non-null affix of order zero, the item is tested in a special tree, since the affix will not be analyzed correctly otherwise. This character position is tested again before the test for reflexivity is carried out, since any verb with a "2" is nonreflexive.

6. Output of the Continuous Dictionary Run

The following sentence from one of the texts in the Harvard tape library will be used to illustrate the output of the Continuous Dictionary Run: "Это флуктуирующее напряжение называется обычно в радиотехнике шумом,

а относительная точность измерения исследуемого напряжения характеризуется величиной отношения напряжения полезного сигнала к среднему квадратичному напряжению шума. Figure 3-20 shows the sentence in texthadic format. The analyzed items are displayed in Fig. 3-21, and the sentence is shown in Fig. 3-22 in final form, after the homographs have been deleted. All the ambiguities that can be resolved by an analysis on a word-by-word basis have been removed. The resolution of the remaining ambiguities is a task left to a more sophisticated program (see Chap. 5).

As a result of the word-by-word analysis, the following information is coded in columns 6 and 7 of the texthadic format (Fig. 3-22): The pronoun "это", the adjective "флуктуирующее", and the noun "напряжение" are neuter and either nominative singular or accusative singular. The adjective, in addition, can function adverbially. The verb "называется" is third person singular, present tense, indicative, and reflexive; while the gender is undetermined. Following it is the short form adjective "обычно", that can function verbally or adverbially. The next word, the preposition "в", governs the accusative or the prepositional case. Next is the essential homograph pair of the noun "радиотехнике", as indicated by the "1" and "%" following the text serial number. The first member of the pair is prepositional singular masculine, while the second member is feminine and either dative or prepositional singular. The next noun, "шумом", is instrumental singular masculine.

After the comma is the conjunction "а", which precedes the adjective "относительная", which is nominative singular feminine. The noun "точность" is either nominative or accusative singular and feminine, and the next one, "измерение", is neuter and either genitive singular, nominative plural, or

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
THIS	P01.00	EHT-O	00A-0080	PKLI STD 0		218923749996
FLUCTUATING	A04.00	FLUKTUIRUJUS HCH-EE	00A-0081	AD0100 4		208798333326
VOLTAGE	N10.00	NAPRJAZHENI- E	00A-0082	NDLIN000		114590000000
TO CALL	V01.00	NAZYVA-ETSJA	00A-0083	VN0OP30000	B0B1B4B6	112270000000
TO BE CALLED	V01.00	NAZYVA-ETSJA	00A-0083	VNROP40000	B0B1B4B6	112280000000
USUAL	A02.00	OBYCHN-O	00A-0084	AD00000		123960000000
IM(TO)	I01.00	-V	00A-0085	R	APOROOBA0650	000020000000
RADIO TECHNI CIAN	N01.10	RADIOTEXNIK- E	00A-0086	NDA1M000		168720000000
RADIO ENGINE ERING	N04.10	RADIOTEXNIK- E	00A-0086	NDI1F100		168730000000
NOISE	A03.00	SHUM-OM	00A-0087	AD00000 1		215880000000
NOISE	N01.00	SHUM-OM	00A-0087	NDI1M000		215870000000
TO MAKE NOIS E	V06.20	SHUM-OM	00A-0087	VN 0000000	B1B4B5B6	215875000000
*.		*.	00A-00880	*		
BUT	I01.00	-A	00A-0089	C		000010000000
RELATIVE(LY)	A02.00	OTNOSITEL,N- AJA	00A-0090	AD0000	T7	132750000000
CONCERNING	I01.00	OTNOSITEL,N- AJA	00A-0090	RH	G00HR0100100	132747500000
ACCURACY	N06.00	TOCHNOST-,	00A-0091	NDI1F1Y0		198730000000
MEASUREMENT	N10.00	IZMERENI-JA	00A-0092	NDI1N000		076520000000
INVESTIGATED	A03.00	ISSLEDUEM-OG O	00A-0093	AD0000 1		083620000000
VOLTAGE	N10.00	NAPRJAZHENI- JA	00A-0094	NDI1N000		114590000000
CHARACTERIZE	V03.00	XARAKTERIZU- ETSJA	00A-0095	VK 0P30000	B1B2B4B5	211140000000
QUANTITY	N04.00	VELICHIN-OJ	00A-0096	NDI2F000		013050000000
RATIO	N10.00	OTNOSHENI-JA	00A-0097	NDI1N000		132780000000
VOLTAGE	N10.00	NAPRJAZHENI- JA	00A-0098	NDI1N000		114590000000
USEFUL	A02.00	POLEZN-OGO	00A-0099	AD00000		149850000000
IT IS USEFUL	I01.00	POLEZN-OGO	00A-0099	X		149845000000
SIGNAL	N01.00	SIGNAL-A	00A-0100	NDI1M000		183370000000
SIGNAL	V04.01	SIGNAL-A	00A-0100	VN OP2L800	B1B4B5	183380000000
TO	I01.00	K-	00A-0101	R	C00R00A00300	084890000000
AVERAGE	A05.00	SREDN-EMU	00A-0102	ADC1000		190910000000
QUADRATIC	A02.00	KVADRATICHN- OMU	00A-0103	AD00000		087170000000
VOLTAGE	N10.00	NAPRJAZHENI- JU	00A-0104	NDI1N000		114590000000
NOISE	A03.00	SHUM-A	00A-0105	AD00000 1		215880000000
NOISE	N01.00	SHUM-A	00A-0105	NDI1M000		215870000000
TO MAKE NOIS E	V06.20	SHUM-A	00A-0105	VN 0000000	B1B4B5B6	215875000000
*.		*.	00A-01060	*		

Sentence from Text after Dictionary Look-up

Fig. 3-20

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
THIS	P01.00	EHT-O	00A-0080	PKLI STD 0	N-A---------- N-N----------		218923749996
FLUCTUATING	A04.00	FLUKTUIRUJUS HCH-EE	00A-0081	AD0100 1 4	N-A---------- N-N----------		208798335326
VOLTAGE	N10.00	NAPRJAZHENI- E	00A-0082	NDI1N000	N-A---------- N-N----------		114590000000
TO CALL	V01.00	NAZYVA-ETSJA	00A-0083	VNOOP30000	INCOMPAT R	B0B1B4B6	112270000000
TO BE CALLED	V01.00	NAZYVA-ETSJA	00A-0083	VNROP40000	--T---BADR	B0B1B4B6	112280000000
USUAL	A02.00	OBYCHN-O	00A-0084	AD00000 2	N------------ N------------		123960000000
IN(TO)	I01.00	-V	00A-0085	P	--A--P--A--P	APOROOBA0650	000020000000
RADIO TECHNI CIAN	N01.10	RADIOTEXNIK- E	00A-0086	NDA1M000	-----P------ -----M-------		168720000000
RADIO ENGINE ERING	N04.10	RADIOTEXNIK- E	00A-0086	NDI1F100	---C-P------ ---F-F-------		168730000000
NOISE	A03.00	SHUM-OM	00A-0087	AD00000 1	INCOMPAT A		215880000000
NOISE	N01.00	SHUM-OM	00A-0087	NDI1M000	----I------- ----M--------		215870000000
TO MAKE NOIS E	V06.20	SHUM-OM	00A-0087	VN 0000000	INCOMPAT A	B1B4B5B6	215875000000
*.		*.	00A-00880	*			
BUT	I01.00	-A	00A-0089	C			000010000000
RELATIVE(LY)	A02.00	OTNOSITEL.N- AJA	00A-0090	AD0000	N------------ F------------	T7	132750000000
CONCERNING	I01.00	OTNOSITEL.N- AJA	00A-0090	PH	INCOMPAT I	G00HR0100100	132747500000
ACCURACY	N06.00	TOCHNOST-.	00A-0091	NDI1F1Y0	N-A---------- F-F----------		198730000000
MEASUREMENT	N10.00	IZMERENI-JA	00A-0092	NDI1N000	-G----N-A--- -N----N-N---		076520000000
INVESTIGATED	A03.00	ISSLEDUEM-OG O	00A-0093	AD0000 1	-GA---------- -BM----------		083620000000
VOLTAGE	N10.00	NAPRJAZHENI- JA	00A-0094	NDI1N000	-G----N-A--- -N----N-N---		114590000000
CHARACTERIZE	V03.00	XARAKTERIZU- ETSJA	00A-0095	VK OP30000	--T---XADR	B1B2B4B5	211140000000
QUANTITY	N04.00	VELICHIN-OJ	00A-0096	NDI2F000	----I------- ----F--------		013050000000
RATIO	N10.00	OTNOSHENI-JA	00A-0097	NDI1N000	-G----N-A--- -N----N-N---		132780000000
VOLTAGE	N10.00	NAPRJAZHENI- JA	00A-0098	NDI1N000	-G----N-A--- -N----N-N---		114590000000
USEFUL	A02.00	POLEZN-OGO	00A-0099	AD00000	-GA---------- -BM----------		149850000000
IT IS USEFUL	I01.00	POLEZN-OGO	00A-0099	X	INCOMPAT I		149845000000
SIGNAL	N01.00	SIGNAL-A	00A-0100	NDI1M000	-G----------- -M-----------		183370000000
SIGNAL	V04.01	SIGNAL-A	00A-0100	VN OP2L600	INCOMPAT A	B1B4B5	183380000000
TO	I01.00	K-	00A-0101	P	---C-----C--	C00R00A00300	084890000000
AVERAGE	A05.00	SREDN-EMU	00A-0102	AD01000	---C--------- ---B---------		190910000000
QUADRATIC	A02.00	KVADRATICHN- OMU	00A-0103	AD00000	---C--------- ---B---------		087170000000
VOLTAGE	N10.00	NAPRJAZHENI- JU	00A-0104	NDI1N000	---C--------- ---N---------		114590000000
NOISE	A03.00	SHUM-A	00A-0105	AD00000 1	INCOMPAT A		215880000000
NOISE	N01.00	SHUM-A	00A-0105	NDI1M000	-G----------- -M-----------		215870000000
TO MAKE NOIS E	V06.20	SHUM-A	00A-0105	VN 0000000	INCOMPAT A	B1B4B5B6	215875000000
*.		*.	00A-01060	*			

Sentence from Text after Analyzer Routines

Fig. 3-21

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
THIS	P01.00	EHT-O	00A-0080	PKLI STD O	N-A----------	N-N----------		218923749996
FLUCTUATING	A04.00	FLUKTUIRUJUS HCH-EE	00A-0081	ADG100 1 4	N-A----------	N-N----------		208798335526
VOLTAGE	N10.00	NAPRJAZHENI- E	COA-0082	NDI1N000	N-A----------	N-N----------		114590000000
TO BE CALLED	V01.00	NAZYVA-ETSJA	00A-0083	VNROP40000	--T---BADR		B0B1B486	112280CC3000
USUAL	A02.00	OBYCHN-O	00A-0084	AD00000 2	N------------	N------------		12396006..000
IN(TO)	I01.00	-V	00A-0085	R	--A--P--A--P		APOROOBAU650	000020000000
RADIO TECHNI CIAN	N01.10	RADIOTEXNIK- E	00A-00861	NDA1M000	-----P------	-----M------		168720006000
RADIO ENGINE ERING	N04.10	RADIOTEXNIK- E	00A-00862	NDI1F100	---C-P------	---F-F------		168730000000
NOISE	N01.00	SHUM-OM	00A-0087	NDI1M000	----I-------	----M-------		215870000000
*.		*;	00A-0088		*			
BUT	I01.00	-A	00A-0089	C				000010000000
RELATIVE(LY)	A02.00	OTNOSITEL.N- AJA	00A-0090	AD0000	N------------	F------------	T7	132750000000
ACCURACY	N06.00	TOCHNOST-.	00A-0091	NDI1FIYO	N-A----------	F-F----------		198730000000
MEASUREMENT	N18.00	IZMERENI-JA	00A-0092	NDI1N000	-G----N-A---	-N----N-N---		076520000000
INVESTIGATED	A03.00	ISSLEDUEM-OG O	00A-0093	AD0000 1	-GA----------	-BM----------		083620000000
VOLTAGE	N10.00	NAPRJAZHENI- JA	00A-0094	NDI1N000	-G----N-A---	-N----N-N---		114590000000
CHARACTERIZE	V03.00	XARAKTERIZU- ETSJA	00A-0095	VX OP30000	--T---XADR		B1B2B485	211140000000
QUANTITY	N04.00	VELICHIN-OJ	00A-0096	NDI2F000	----I-------	----F-------		013050000000
RATIO	N18.00	OTNOSHENI-JA	00A-0097	NDI1N000	-G----N-A---	-N----N-N---		132780000000
VOLTAGE	N10.00	NAPRJAZHENI- JA	00A-0098	NDI1N000	-G----N-A---	-N----N-N---		114590000000
USEFUL	A02.00	POLEZN-OGO	00A-0099	AD00000	-GA----------	-BM----------		149850000000
SIGNAL	N01.00	SIGNAL-A	00A-0100	NDI1M000	-G-----------	-M-----------		183370000000
TO	I01.00	K-	00A-0101	R	---C-----C--		COOROOA00300	084890000000
AVERAGE	A05.00	SREDN-EMU	00A-0102	AD01000	---C---------	---B---------		190910000000
QUADRATIC	A02.00	KVADRATICHN- OMU	00A-0103	AD00000	---C---------	---B---------		087170000000
VOLTAGE	N10.00	NAPRJAZHENI- JU	00A-0104	NDI1N000	---C---------	---N---------		114590000000
NOISE	N01.00	SHUM-A	00A-0105	NDI1M000	-G-----------	-M-----------		215870000000
*.		*.	00A-0106		*			

Augmented Text of Sample Sentence

Fig. 3-22

accusative plural. The adjective "исследуемого" is genitive or accusative singular. If genitive, it can be masculine or neuter; but if accusative, it can only be masculine. The coding for the noun "напряжение" is like that for "измерение". The verb "характеризуется" is third person singular, either present or future tense, indicative, reflexive, and the gender is undetermined. The following noun, "величиной", is instrumental singular feminine.

The coding for the noun "отношение" is similar to that for "напряжение", which has been described previously, while that for the adjective "полезного" is similar to that for "исследуемого". The noun "сигнала" is genitive singular masculine. The preposition "к", which governs the dative case, follows, preceding the two adjectives "среднему" and "квадратичному", which are dative singular and either masculine or neuter. The next noun, "напряжению", is dative singular neuter, and the last word, the noun "шума", is genitive singular masculine.

Of the seven homograph sets† contained in the sentence, six were resolved by the analyzer programs as follows (Fig. 3-22):

The two dictionary entries for "называется" differ in the third character position of the organized word. One entry was intended for reflexive forms of the verb and the other entry for nonreflexive forms.

The homographic pair "радиотехнике" is an essential homograph. Since the homograph cannot be resolved without a consideration of context, its resolution is left to a future program.

---

†A <u>homograph set</u> consists of two or more dictionary entries, looked up by the same inflected form, that are successfully analyzed by the analyzer programs.

There are two sets of three homographs referring to the same dictionary stems, "шумом" and "шума". In both cases, both the adjectival and verbal stems are incompatible, leaving only a single compatible nominal entry. The adjectival stem is an example of a stem automatically marked by the anomalous stem routine (note the "1" in character position 12 of column 5).

The next homographic pair is resolved, since the indeclinable dictionary entry refers only to "относительн-о" and not to "относительн-ая". The next homographic pair is resolved in the same manner, the indeclinable entry referring only to "полезн-о" and not to "полезн-ого".

The verbal entry for "сигнала" is rejected, since the affix "a" in a verb is an indication of a past tense and there is no signal in column 8 that a past tense (B3) can occur with the stem "сигнал".

7. Reliability of the Harvard Automatic Dictionary

The reliability of the Harvard Automatic Dictionary and of the look-up routines constituting the Continuous Dictionary Run is tested periodically by means of the output of Frequency Runs.[22] A list, containing every distinct inflected form from every text in the Harvard tape library, together with the frequency of occurrence of each form, is kept on tape. (Ref. 23 contains a list of all texts in the tape library.) The latest test, Frequency Run V, processed in January 1960, was based on 107,097 words of text consisting of 14,698 distinct inflected forms.

A selection from the output of the latest test run is shown in Fig. 3-23. Several items of special interest that appear on this excerpt

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)		FREQUENCY OF OCCURRENCE	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
LOCATION	N10.00	MESTONAXOZHD ENI-JA		0001	NDI1N000	-G----N-A--- -N----N-N---		106166666666
POSITION	N10.00	MESTOPOLOZHE NI-E		0001	NDI1N000	N-A---------- N-M----------		106173333333
LAYER	N10.00	MESTOROZHDEN I-J		0001	NDI1N000	--------G---- -------N----		106176666666
HERE AND THE RE	I01.00	MEST-U		0001	H	INCOMPAT I		106130000000
PLACE	N08.00	MEST-U		0001	NDI1N000	---C--------- ---N--------		106140000000
METALLIC	A03.00	METALL-A	HOMOGRAPH	0002	AD00000 1	N------------ F-----------		106235000000
METAL	N01.00	METALL-A		0002	NDI1M000	-G----------- -M----------		106230000000
METALLIC	A03.00	METALL-AX		0002	AD00000	INCOMPAT A		106235000000
METAL	N01.00	METALL-AX		0002	NDI1M000	------------P -----------M		106230000000
METALLIC	A03.00	METALL-E		0001	AD00000	INCOMPAT A		106235000000
METAL	N01.00	METALL-E		0001	NDI1M000	-----P------ -----M------		106230000000
METAL	A06.00	METALLICHESK -IJ		0001	AD01000	N-A--------- M-M---------		106460000000
METAL	A06.00	METALLICHESK -IMI		0001	AD01000	-----------I- ----------A-		106460000000
METAL	A06.00	METALLICHESK -IX		0001	AD01000	--------GA--P -------AA--A		106460000000
METAL	A06.00	METALLICHESK -OJ		0002	AD01000	-G-CIP------ -F-FFF------		106460000000
METAL	A06.00	METALLICHESK -UJU		0002	AD01000	--A---------- --F---------		106460000000
METALLICHESL -IX		METALLICHESO -IX	MISSPELLING	0001	***			
METALLIC	A03.00	METALL-O		0008	AD00000 2	N------------ N-----------		106235000000
METAL	N01.00	METALL-O		0008	NDI1M000	INCOMPAT A		106230000000
METALLIC	A03.00	METALL-OV		0001	AD00000	INCOMPAT A		106235000000
METAL	N01.00	METALL-OV		0001	NDI1M000	-------G---- -------M----		106230000000
(SEMANTIC BL ANK)+	A01.00	METALLOSTEKL JANN-OJ	MISSING WORD	0002	AD0000	-G-CIP------ -F-FFF------	Q3	PSEUDOENTRY
METALLURGY	N07.00	METALLURGI-I		0001	NDI1F100	-G-C-PN-A--- -F-F-FF-F---		106490000000
(SEMANTIC BL ANK)+	A06.00	METALLURGICH ESK-OJ		0001	AD0000	-G-CIP------ -F-FFF------		PSEUDOENTRY
METALLIC	A03.00	METALL-Y	HOMOGRAPH	0002	AD00000 1	------N----- ------A-----		106235000000
METAL	N01.00	METALL-Y		0002	NDI1M000	------N-A--- ------M-M---		106230000000
METHANE	N01.00	METAN-A		0001	NDI1M100	-G---------- -M----------		106496666666
METEORIC	A02.00	METEORN-OGO		0002	AD01000	-GA--------- -BM---------		106540000000
METEORIC	A02.00	METEORN-YMI		0001	AD01000	----------I- ----------A-		106540000000
METEORIC	A02.00	METEORN-YX		0003	AD01000	--------GA--P -------AA--A		106540000000
METEOR-OV		METEOR-OV	MISSING WORD	0007	***			
METEOROLOGIC AL	A06.00	METEOROLOGIC HESK-IX		0001	AD00000	--------GA--P -------AA--A		106583333333
METEOR-Y		METEOR-Y		0001	***			
MARKED	A03.00	MET-OV	PROBLEM SET	0001	AD0000 1 1	INCOMPAT A		106180000000
MARK	V04.01	MET-OV		0001	VN OP700KO	INCOMPAT A	C3 B185	106190000000
METHOD	N01.00	METOD-		0035	NDI1M000	N-A---------- M-M----------		106660000000
METHOD	N01.00	METOD-A		0012	NDI1M000	-G----------- -M----------		106660000000
METHOD	N01.00	METOD-AM		0006	NDI1M000	---------C-- ---------M--		106660000000
METHOD	N01.00	METOD-AMI		0006	NDI1M000	----------I- ----------M-		106660000000
METHOD	N01.00	METOD-AX		0004	NDI1M000	------------P ------------M		106660000000
METHOD	N01.00	METOD-E		0005	NDI1M000	-----P------ -----M------		106660000000
METHOD	N04.10	METODIK-A		0013	NDI1F000	N------------ F-----------		106670000000
METHOD	N04.10	METODIK-E		0001	NDI1F000	---C-P------ ---F-F------		106670000000
METHOD	N04.10	METODIK-I		0002	NDI1F000	-G----N-A--- -F----F-F---		106670000000
METHOD	N04.10	METODIK-U		0001	NDI1F000	--A---------- --F---------		106670000000
METHOD	N01.00	METOD-OV		0018	NDI1M000	--------G---- -------M----		106660000000

Sample of Main Output of Frequency Run V

Fig. 3-23

are two homograph sets, "металл-а" and "металл-ы"; a problem set,* "мет-ов"; a misspelled word, "металличеол-их"; as well as four words that were missing from the dictionary. Two of these missing words have been analyzed by the missing word analyzer,[21] "металлостеклянн-ой" and "металлургическ-ой", while the other two, "метеор-ов" and "метеор-ы", could not be analyzed by that routine and are listed as missing words.

To find errors in the output, three supplementary lists were produced: a list of homograph sets (Fig. 3-24), a list of problem sets (Fig. 3-25), and a list of all the incompatible items from the main output sorted by class (Fig. 3-26).

Only a single error was noted on the list of incompatible items. This error was noted again on the list of problem sets. The information gleaned from the homograph set list and the problem set list is summarized in Tables 3-13 and 3-14. The data refers to the distinct inflected forms as well as to the text occurrences, so that a clear picture of the magnitude of errors in the dictionary and the associated routines can be discerned.

The homographs that were found in the output of Frequency Run V have been classified into six groups. The first and by far the largest group consists of the essential, or genuine, homograph sets. One member of every homograph set in the second group is a short form adjective whose existence is questionable but which has been left in the dictionary, since there is as yet no reliable source of information on this subject.

The homograph sets in the third group are due to duplicate entries in the dictionary, whereas those in the fourth group are caused by coding

---

* A problem set consists of one or more dictionary entries which have been looked up by the same inflected form, and which all have been identified as incompatible items by the analyzer programs.

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)		FREQUENCY OF OCCURRENCE	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
WING	N99.99	KRYL-O	(5)	0001	NDI2N000	INCOMPAT Z			097538750000
COVERED	V14.00	KRYL-O		0001	VNOOP30000	SSS---AND-		B3	097536250000
RELATIVE BEA RING	A03.00	KURS-OV	(2)	0001	AD00000 1 1	N------------	M-----------		098100000000
COURSE	N01.00	KURS-OV		0001	NDI1M000	--------G----	--------M----		098090000000
RAY	N03.10	LUCH-A	(4)	0003	NDI1M000	-G-----------	-M-----------		101610000000
SHINE BRIGHT LY	V04.20	LUCH-A		0003	VN 3000000	B G-		B1B4B5	101630000000
RAY	N03.10	LUCH-I		0004	NDI1M000	-------N-A---	-------M-M---		101610000000
SHINE BRIGHT LY	V04.20	LUCH-I		0004	VN 3000000	-Z---- E-		B1B4B5	101630000000
RAY	N03.10	LUCH-U		0001	NDI1M000	---C---------	---M---------		101610000000
SHINE BRIGHT LY	V04.20	LUCH-U		0001	VN 3000000	V-----BAD-		B1B4B5	101630000000
BETTER	A04.80	LUCHSH-E	(2)	0002	AD00000 2	N------------	N-----------		101865000000
BETTER	I01.00	LUCHSH-E		0002	H				101860000000
LITTLE	D01.80	MAL-O	(1)	0008	D XEACUNYSS	N-A---N-A---	A-A---A-A---	0T0000T0000	102968335336
SMALL	A05.00	MAL-O		0008	KDA1000 2	N------------	N-----------		102970000000
MATHEMATICIA N	N01.10	MATEMATIK-		0001	NDA1M000	N------------	M-----------		104711000000
MATHEMATICS	N04.10	MATEMATIK-		0001	NDI1F100	--------G----	--------F----		104712000000
MATHEMATICIA N	N01.10	MATEMATIK-A		0004	NDA1M000	-GA----------	-MM----------		104711000000
MATHEMATICS	N04.10	MATEMATIK-A		0004	NDI1F100	N------------	F-----------		104712000000
MATHEMATICIA N	N01.10	MATEMATIK-AM I		0001	NDA1M000	-----------I-	-----------M-		104711000000
MATHEMATICS	N04.10	MATEMATIK-AM I		0001	NDI1F100	-----------I-	-----------F-		104712000000
MATHEMATICIA N	N01.10	MATEMATIK-I		0006	NDA1M000	-------N-----	-------M-----		104711000000
MATHEMATICS	N04.10	MATEMATIK-I		0006	NDI1F100	-G----N-A---	-F----F-F---		104712000000
HERE AND THE RE	I01.00	MEST-AMI		0006	H				106130000000
PLACE	N08.00	MEST-AMI		0006	NDI1N000	-----------I-	-----------N-		106140000000
METALLIC	A05.00	METALL-A	(2)	0002	AD00000 1	N------------	F-----------		106235000000
METAL	N01.00	METALL-A		0002	NDI1M000	-G-----------	-M-----------		106230000000
METALLIC	A03.00	METALL-Y		0002	AD00000 1	-------N-----	-------A-----		106235000000
METAL	N01.00	METALL-Y		0002	NDI1M000	-------N-A---	-------M-M---		106230000000
MECHANICS	N04.10	MEXANIK-A	(1)	0006	NDI1F000	N------------	F-----------		106792500000
ENGINEER	N01.10	MEXANIK-A		0006	NDI1M000	-G-----------	-M-----------		106790000000
MECHANICS	N04.10	MEXANIK-AMI		0001	NDI1F000	-----------I-	-----------F-		106792500000
ENGINEER	N01.10	MEXANIK-AMI		0001	NDI1M000	-----------I-	-----------M-		106790000000
MECHANICS	N04.10	MEXANIK-E		0007	NDI1F000	---C-P------	---F-F------		106792500000
ENGINEER	N01.10	MEXANIK-E		0007	NDI1M000	-----P------	-----M------		106790000000
MECHANICS	N04.10	MEXANIK-I		0006	NDI1F000	-G----N-A---	-F----F-F---		106792500000
ENGINEER	N01.10	MEXANIK-I		0006	NDI1M000	-------N-A---	-------M-M---		106790000000
TO MEET	I01.00	NAVSTRECH-U		0001	H				111655000000
COMING FROM THE OP	I01.80	NAVSTRECH-U		0001	RH	--C-------C--		C00HR0000000	111655625000
HEATING	N01.00	NAGR-EV		0005	NDI1M100 1	N-A----------	M-M----------		111680000000
WARM	V01.00	NAGR-EV		0005	VS OP30000	A G-		B4B6	111690000000
ONE MUST	I01.00	NAD-O		0014	H				111927500000
ABOVE	I01.00	NAD-O		0014	P	----I-----I-		I00R00500100	111928125000
SIDE BY SIDE	I01.00	NARJAD-U		0004	H				115150000000
ORDER	N01.00	NARJAD-U		0004	NDI1M000	---C---------	---M---------		115140000000
WAS	V04.00	NAXODIL-AS,	(3)	0001	VN 0000000	SSS---AFDR		B3	115802027025
FOUND	V04.00	NAXODIL-AS,		0001	VN OP300KO	SSS---AFDR	E5	B3	115802162160
WAS	V04.00	NAXODIL-IS,		0001	VN 0000000	---PPPAADR		B3	115802027025
FOUND	V04.00	NAXODIL-IS,		0001	VN OP300KO	---PPPAADR	E5	B3	115802162160

Section from Homograph Set List, Frequency Run V

Fig. 3-24

```
IMENN-JU                A01.00 X00   XXX-34740001 NAMED       *

IMENN-JU                0001   T10   INCOMPAT A   078635000000 AD00000      A0
                        I01.00 X00   XXX-34740001 NAMELY      1 TO WIT    2 JUST      3 EXACTLY    *

                        0001   G00   INCOMPAT I   078630000000 H            A0


IN-T-A                  I01.00 100   XXX-36640002 INSTITUTF   *

                        0002         INCOMPAT I   080985000000 S            A6



ISKRENN-JUJU            A01.00 XX0   XXX-37130001 SPARKED     1 FLASHED    *

ISKRENN-JUJU            0001   T10   INCOMPAT A   081980000000 AD00C0    3  A1
                        A01.00 XX0   XXX-37130001 SINCERE     1 CANDID     *

                        0001   T10   INCOMPAT A   081985000000 AD00000      A0


KVADRAT-O               N01.00 G00   XXX-40460001 SQUARE      *

                        0001   100   INCOMPAT A   087140000000 NDI1M000     A0



KOL-,                   I01.00 U00   XXX-41960001 IF          *

KOL-,                   0001   900   INCOMPAT I   089710000000 H            A0
                        V11.10 U00   XXX-41960001 TO BREAK    1 TO THRUST 2 TO PRICK  3 TO STAB    4 TO CHOP    *

                        0001   X00   INCOMPAT A   089715000000 VN OP30000   A0                       M1B485B6
```

Problem Sets from Frequency Run V

Fig. 3-25

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	FREQUENCY OF OCCURRENCE	ORGANIZED WORD		CODING DUE TO WORD-BY-WORD ANALYZER	DICTIONARY SERIAL NO.
ORGANIZATION	N07.00	ORGANIZATSI-: AMI MISSPELLING	0001	NDI1F000		INCOMPAT A	128530000000
AFFAIR	N08.00	DEL-IM	0001	NDI1N000		INCOMPAT A	048510000000
AFFAIR	N08.00	DEL-ITSJA VERB	0007	NDI1N00J		INCOMPAT A	048510000000
AFFAIR	N08.00	DEL-JA	0001	NDI1N000		INCOMPAT A	048510000000
AFFAIR	N08.00	DEL-JATSJA	0004	NDI1N000		INCOMPAT A	048510000000
BOTTOM	N08.00	DN-EGO	0001	NDI1N000		INCOMPAT A	052060000000
BOTTOM	N08.00	DN-EJ PRONOUN	0001	NDI1N000		INCOMPAT A	052060000000
CIRCUMSTANCE	N10.00	OBSTOJATEL,S TV-O WRONG CLASS: N8	0010	NDI1N000		INCOMPAT A	123410937500
CIRCUMSTANCE	N10.00	OBSTOJATEL,S TV-A	0002	NDI1N000		INCOMPAT A	123410937500
DIFFERENCE	N10.00	OTLICHI-T, VERB	0002	NDI1N000		INCOMPAT A	132367500000
STRETCHING	N10.00	RASTJAZHENI- A MISSPELLING	0001	NDI1N000		INCOMPAT A	174640000000
COMPUTING	N10.00	RESHENI-V SHORT FORM ADJECTIVE	0001	NDI1N000		INCOMPAT A	177890000000
CONDITION	N10.00	USLOVI-AX MISSPELLING	0001	NDI1NOYO		INCOMPAT A	205630000000
CONDITION	N10.00	USLOVI-VSHIS VERB	0005	NDI1NOYO		INCOMPAT A	205630000000
FIELD	N13.00	POL-ETE NI	0001	NDI1N000		INCOMPAT A	149720000000
MODEL	I01.00	TABL-I NOUN	0001	NUION000		INCOMPAT I	195170000000
DASH	I01.00	TIR-OV MISSING VERB	0001	NUION000		INCOMPAT I	197800000000
VIDEO	I01.00	VIDE-T, VERB	0056	NUION000		INCOMPAT I	016070000000
TREE	N99.99	DEREV-A	0023	NDI2N000		INCOMPAT Z	049220769230
TREE	N99.99	DEREV-O	0005	NDI2N000		INCOMPAT Z	049220769230
TREE	N99.99	DEREV-OM	0002	NDI2N000		INCOMPAT Z	049220769230
TREE	N99.99	DEREV-U	0001	NDI2N000		INCOMPAT Z	049220769230
TREE	N99.99	DEREV,-EV	0005	NDI2N000		INCOMPAT Z	049221538460
TREE	N99.99	DEREV,-JA	0001	NDI2N000		INCOMPAT Z	049221538460
WING	N99.99	KRYL-A	0003	NDI2N000		INCOMPAT Z	097538750000
WING	N99.99	KRYL-O N99.99	0001	NDI2N000		INCOMPAT Z	097538750000
LINK	N99.99	ZVEN-A	0005	NDI2N000		INCOMPAT Z	071275625000
LINK	N99.99	ZVEN-OM	0001	NDI2N000		INCOMPAT Z	071275625000
LINK	N99.99	ZVEN-U	0001	NDI2N000		INCOMPAT Z	071275625000
LINK	N99.99	ZVEN,-EV	0005	NDI2N000		INCOMPAT Z	071278750000
LINK	N99.99	ZVEN,-JA	0002	NDI2N000		INCOMPAT Z	071278750000
LINK	N99.99	ZVEN,-JAMI	0004	NDI2N000		INCOMPAT Z	071278750000
LINK	N99.99	ZVEN,-JAX	0001	NDI2N000		INCOMPAT Z	071278750000
APOHM	N01.00	AB- АБОМ	0001	NDI1M000	1	INCOMPAT A	000185000000
APOHM	N01.00	AB-	0001	NDI1M000	1	INCOMPAT A	000180000000
ADDRESSEE	N01.00	ADRES-	0010	NDA1M000	1	INCOMPAT A	001290000000
ADDRESSEE	N01.00	ADRES-A	0018	NDA1M000	1	INCOMPAT A	001290000000
ADDRESSEE	N01.00	ADRES-AMI АДРЕСАТ	0002	NDA1M000	1	INCOMPAT A	001290000000
ADDRESSEE	N01.00	ADRES-AX	0002	NDA1M000	1	INCOMPAT A	001290000000
ADDRESSEE	N01.00	ADRES-E	0008	NDA1M000	1	INCOMPAT A	001290000000
ADDRESSEE	N01.00	ADRES-OV	0005	NDA1M000	1	INCOMPAT A	001290000000
ADDRESSEE	N01.00	ADRES-U	0003	NDA1M000	1	INCOMPAT A	001290000000

Section from Incompatible List, Frequency Run V

Fig. 3-26

	Distinct Inflected Forms		Text Occurrences	
1. Essential homographs	165	46%	4214	84%
2. Short form adjectives	83	23	360	7
3. Duplicates in dictionary	61	17	254	5
4. Dictionary coding errors	44	12	156	3
5. Words not in classes	4	1	6	-
6. Analyzer errors	1	-	15	-
	358		5005	

358 out of 14,698 distinct inflected forms (2.4%)

5,005 out of 104,097 words of text (4.8%)

Summary of Homograph Set List, Frequency Run V, January 1960

TABLE 3-13

	Distinct Inflected Forms		Text Occurrences	
1. Words missing from dictionary	62	41%	203	56%
2. Typographical errors	63	42	72	20
3. Dictionary coding errors	23	15	80	22
4. Analyzer errors	2	1	9	2
	150		364	

150 out of 14,698 distinct inflected forms (1.0%)

364 out of 104,097 words of text (0.3%)

Summary of Problem Sets, Frequency Run V, January 1960

TABLE 3-14

errors in the dictionary. Homograph sets originating from words that cannot be classified into normal classes (words with class markers greater than 75) are listed in the fifth group, while the last group is reserved for homograph sets caused by errors in the analyzer programs.

While the homographs in groups 1, 2, and 5 are considered essential at present, the errors that caused the other homograph sets have been corrected.

Examples of homograph sets belonging to the first five groups may be found in Fig. 3-24. The pertinent groups have been marked to the right of the column containing the transliterated Russian word. The assignment of the homograph sets to the six groups is self-evident, perhaps with the exception of the homograph sets with the verb stem "луч". This verb can exist only in the reflexive voice, but the dictionary entry was not appropriately marked in character position 3 of the organized word.

The data of Table 3-13 indicates that almost 5% of the words occurring in the texts on the Harvard tape library refer to homographic dictionary entries. Although any given homograph set is a function of the morphological classes that have been assigned to the individual members of the set, and in that manner a function of the organization of the Harvard Automatic Dictionary, the latitude allowed the coders is not great. It is therefore likely that any other automatic dictionary would have to be capable of handling homograph sets that occur with approximately the same frequency.

In the present dictionary, fewer than 0.5% of the words in texts refer to homograph sets due to errors.

The problem sets have also been classified into groups (Table 3-14). The first group consists of problem sets created by the absence of a text word

from the dictionary. If the stem of the text word is homographic with the stem of another Russian word represented in the dictionary and subsequently rejected by the analyzer programs, the problem set occurs. It is important to note that not every text word missing from the dictionary results in a problem set. The majority of words missing from the dictionary is, of course, not homographic with the stem of another word. New words not homographic with other stems are listed as missing words with no English correspondents and no grammar codes, unless such codes can be assigned by the missing word analyzer.

Another group of problem sets is due to typographical errors, generated when the text is being typed onto a magnetic tape. Here, too, not every word typed erroneously results in a problem set. Most appear as missing words. A mistyped word can result in a problem set only in one of two circumstances. Either the typographical error is in the affix and the analyzer program cannot correlate the incorrect affix with the stem, or the error is in a stem which coincidentally is identical to the stem of another dictionary word.

The other two groups of problem sets are due to dictionary coding errors and errors in the analyzer programs. All such errors discovered through reference to the homograph list and the problem set list have been corrected.

Examples of the first three types of problem sets are illustrated in Fig. 3-25. The word коль, an alternate form of коли*, is missing from the dictionary, but is homographic with the same stem from the forms "коли" and "колю", the latter from the paradigm of колоть*. Two misspellings are on the list: "именно" was spelled "именню" and "квадрата" was spelled

"квадрато". The other two examples are due to dictionary errors. The adjective мокремний* (form "мокремнию") was misclassified into class A1 instead of A5, while the abbreviation мп-т* was listed as being indeclinable when it can be declined, as "мп-та".

Problem sets are created by text words extremely rarely (0.3%), and those due to dictionary errors occur even more seldom (less than 0.1% of the time).

8. Frequency of Occurrences of Affixes

Since the three word analyzer programs are used to analyze every Russian word of the noun, adjective, or verb morphological types, it was desirable to resolve several statistical questions in order to reduce the time involved in passing through the logical trees of these three programs.

In the main branch of each program the affix of the text word is compared against a list of affixes stored in memory. If the affix lists are stored in order of decreasing frequency of occurrence, the least time will be spent passing through the trees. Since the data that is processed by the analyzer programs is the raw output of dictionary look-up, the statistics should reflect the frequency of occurrence of all 30-word dictionary items, both compatible and incompatible.

Frequency Run V has already been considered in Sec. 7, where the individual entries have been studied for indication of error. This data also has been reduced to obtain the desired frequencies.

Every 30-word item in the analyzer output is compressed until only the morphological type, affix, class marker, an index whether the item is compatible or incompatible, and the frequency of occurrence of the item are

kept. This information is sorted and then accumulated (Table 1 of Appendix E). In the table the three keys, in decreasing order, are the assigned morphological type, the affix, and the class marker. The totals for each summation are divided into compatible and incompatible items. The totals for the affixes within the major morphological types have been sorted by frequency of occurrence (Tables 2 to 4 of Appendix E). This is the order in which the affixes must be listed in the analyzer programs to reduce the scanning time.

The figures in Table 1 in Appendix E have been summarized further in Table 3-15. It must be noted that there is not a one-to-one correspondence between the figures in Sec. 7 and those of Table 3-15, since a distinct inflected form may refer to more than one dictionary entry.

Morphological Type	Total Entries	Compatible	Incompatible
Noun	35,875	33,030	2,845
Indeclinable	32,166	27,271	4,895
Adjective	24,312	18,807	5,505
Verb	22,265	10,200	12,065
Pronoun	8,225	8,223	2
Numeral	1,381	1,276	105
	124,224	98,807	25,417
		(79.5%)	(20.5%)
Miscellaneous	30,012		
	154,236		

Summary of Dictionary Entries Looked Up in Frequency Run V

TABLE 3-15

The reason for selecting the analyzer programs and other related procedures as the method for determining the compatibility and lexical attributes of the various word types was based on the development of the existing experimental system. The reduction in efficiency due to this method can be determined by studying the ratio of incompatible items that have to be carried through up to the homograph delete routine in the Continuous Dictionary Run (Fig. 3-1). The 20.5% ratio is an indication of the useless data being carried through the several routines. The necessity for this could be eliminated by more efficient coding procedures and a larger internal memory.

The difficulties caused by the large number of dictionary stems in each verb paradigm are pointed out by the statistic that almost half of the incompatible items are verbs. The large number of stems are a result of the affixes factored by the inverse inflection algorithm (Sec. 2B).

The 30,012 miscellaneous items that are appended to the main list include punctuation marks, editorial comments made by typists during text transcription, and words that were not found in the dictionary. A rough estimate of the number of missing words is 5,000. The missing words include many proper names and most of the typographical errors generated during transcription.

REFERENCES ⁎

1. Giuliano, V. E., "An Experimental Study of Automatic Language Translation," Doctoral Thesis, Harvard University (1959).

2. Report No. NSF-2 (1959).

3. Report No. NSF-3 (1959).

4. Report No. NSF-4 (1960).

5. Jones, P. E., Jr., "Modifications to the Continuous Dictionary Run (I)," NSF-3, Sec. XVI (1959).

6. Jones, P. E., Jr., "The Continuous Dictionary Run," NSF-2, Sec. I (1959).

7. Oettinger, A. G., <u>Automatic Language Translation: Lexical and Technical Aspects</u>, Harvard University Press, Cambridge, Mass. (1960).

8. Magassy, K., "An Automatic Method of Inflection for Russian," AF-46, Sec. V (1957)..

9. Magassy, K., "An Automatic Method of Inflection for Russian (II)," AF-49, Sec. III (1957).

10. Matejka, L., "Grammatical Specifications in the Russian-English Automatic Dictionary," AF-50, Sec. V (1958).

11. Matejka, L., "The Automatic Interpretation of Russian Verbal Endings," NSF-2, Sec. III (1959).

12. Foust, W. H., "Inflected Form Generators," AF-49, Sec. VI (1957).

13. Foust, W. H., "Inflectors," AF-50, Sec. VI (1958).

⁎ Throughout this Reference List the following abbreviations will be used:

AF-49, 50, etc. – <u>Design and Operation of Digital Calculating Machinery</u>, Progress Reports by the Staff of the Computation Laboratory of Harvard University to the United States Air Force, Cambridge, Mass.

NSF-2, 3, etc. – <u>Mathematical Linguistics and Automatic Translation</u>, Report to the National Science Foundation, The Computation Laboratory of Harvard University, Cambridge, Mass.

14. Oettinger, A. G., "A Study for the Design of an Automatic Dictionary," Doctoral Thesis, Harvard University (1954).

15. Frink, O., "Modifications to the Programs for Correcting the Harvard Automatic Dictionary and for Syntactic Study," NSF-3, Sec. IV (1959).

16. Frink, O., "Programs for Correcting the Harvard Automatic Dictionary and for Syntactic Study (Conhadic, Checkhadic, Texthadic, and Freqhadic), NSF-2, Sec. V (1959).

17. Matejka, L., "Grammatical Coding for Pronouns," NSF-3, Sec. XVIII (1959).

18. Coppinger, L. and von Susich, S., "Grammatical Coding," NSF-4, Sec. III (1960).

19. Magassy, K., "Russian Numerals in the Harvard Dictionary File," NSF-4, Sec. IV (1960).

20. Matejka, L., "Grammatical Coding for Prepositions," NSF-3, Sec. VI (1959).

21. Jones, P. E., Jr., "Modifications to the Continuous Dictionary Run (II)," NSF-4, Sec. XIII (1960).

22. von Susich, S., "Frequency Runs - A System for Lexical Quality Control and Statistical Analysis," NSF-3, Sec. VIII (1959).

23. Coppinger, L., "Bibliography of Recorded Russian Texts," NSF-4, Sec. II (1960).

CHAPTER 4

A MODEL FOR NATURAL LANGUAGE

1. Introduction

It is helpful to construct a theoretical foundation to explain the important features of a predictive syntactic analysis technique for the Russian language, empirically devised by Rhodes[1] and adopted with modifications at Harvard University (see Chapter 5). A working model of natural language that can be analyzed by this technique is presented in this chapter. This model is based on the formalization of the syntax of Łukasiewicz' parenthesis-free notation given by Burks, Warren, and Wright,[2] on the linguistic model of Chomsky,[3,4] and on Oettinger's theory of syntactic analysis.[5] This theory utilizes a storage device consisting of a linear array of storage elements, in which information is entered and removed from one end only in accordance with a "last-in-first-out" principle. Among programmers this storage device has come to be known as a pushdown store. The importance of the pushdown store for a similar analysis was recognized independently by Samelson and Bauer.[6] Familiarity with the Burks, Warren, and Wright paper is assumed in this chapter.

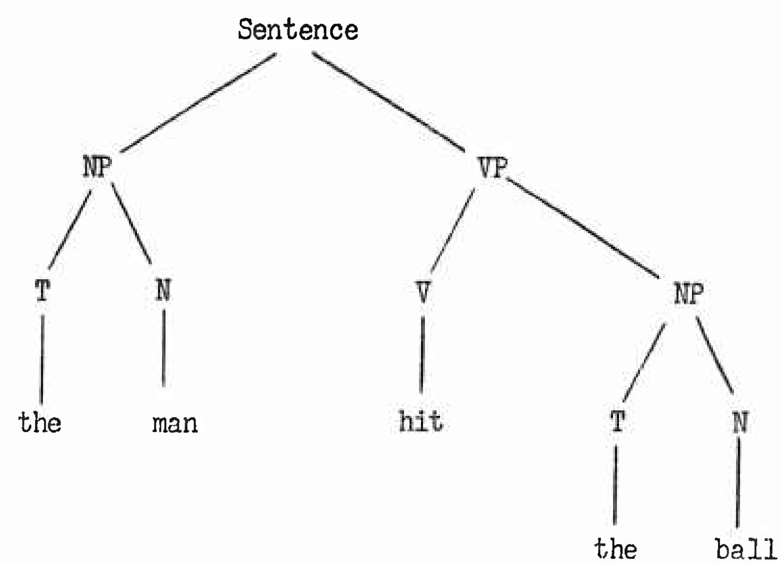
The technique of predictive syntactic analysis is based on the observation that in scanning a Russian sentence from left to right, it is possible, on the one hand, to make predictions about the syntactic structures that occur further to the right, and on the other hand, to determine the syntactic role of the word currently being examined by testing it against the previously made predictions that it might fulfill. The predictions are

stored in a prediction pool, a device with characteristics approximately those of a simple pushdown store, as described by Oettinger. Predictions are tested for fulfillment downward from the top of the prediction pool, but new predictions are always entered at the top of the pool.

In his phrase structure model for the synthesis of English sentences, Chomsky has related the syntactic roles of the words in a sentence to each other by a hierarchy of grammatical rules expressed in the form

$$X_i \longrightarrow Y_i,$$

where Y_i is formed from X_i by the replacement of a single symbol of X_i by some string of one or more symbols. The vocabulary that characterizes the terminal strings is the set of English words of the sentence being synthesized (Fig. 4-1). The rules for the derivation of the sample sentence of Fig. 4-1 are given in Table 4-1.



Derivation of the Sentence: "The man hit the ball".

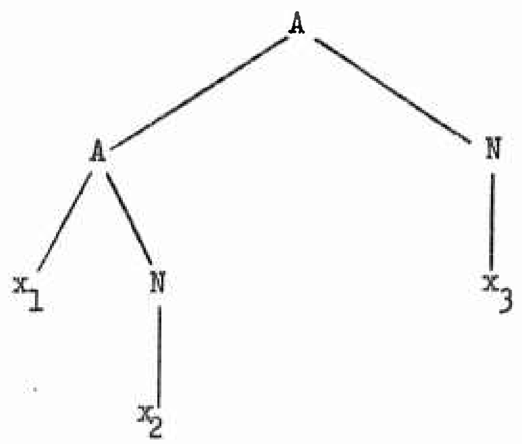
Fig. 4-1

Sentence $\longrightarrow$ NP + VP	NP - noun phrase
NP $\longrightarrow$ T + N	VP - verb phrase
VP $\longrightarrow$ V + NP	T - article
T $\longrightarrow$ the	N - noun
N $\longrightarrow$ man, ball	V - verb
V $\longrightarrow$ hit	

Rules for the Derivation of the Sentence: "The man hit the ball".

TABLE 4-1

A statement in the Łukasiewicz' parenthesis-free notation, as described by Burks, Warren and Wright, can be represented by a tree-like structure, paralleling Chomsky's representation for sentence synthesis. In the illustration (Fig. 4-2) three different types of characters are used: the monadic functor N, the dyadic functor A, and the variables x_i.



Representation of the Formula $\Delta = AAx_1\ Nx_2\ Nx_3$.

Fig. 4-2

The set of functors in the parenthesis-free notation is analogous to the set of characters, such as "NP", "VP", "N", etc., in the intermediate language of phrase structure; the set of variables in the parenthesis-free

notation is analogous to the set of characters, the English words, in the terminal language.

Oettinger's syntactic analysis theory is based on the proof of a "Δ_M-theorem" for the algorithms that he has proposed. Let Δ, which represents any formula in the universe of formulas to be analyzed, be split into a head Δ_H, a middle Δ_M, and a tail Δ_T, such that $\Delta = \Delta_H\Delta_M\Delta_T$. Δ_M is assumed to be "well-formed", while Δ_H and Δ_T are arbitrary residues determined by the choice of Δ_M. The theorem states that if, at a certain point in the left-to-right syntactic analysis of Δ, (1) Δ_H has been analyzed, (2) the output of the analysis is a function of Δ_H, and (3) the content of the pushdown store is a function of Δ_H only, then at a later point, after Δ_M has been analyzed, the output will be a function of both Δ_H and Δ_M, but the pushdown store still will be the function of Δ_H as in condition (3).

Oettinger has defined a set of three parenthetic notations: the familiar full parenthetic notation, a left parenthetic notation in which all the right parentheses have been removed from the full parenthetic notation, and a right parenthetic notation in which all the left parentheses have been removed from the full parenthetic notation (Fig. 4-3). With the Δ_M-theorem, he has shown the feasibility of translating between the parenthesis-free notation and any one of the several alternative parenthetic notations. The translation algorithms, which also yield syntactic analyses of the formulas, have the following interesting properties:

1. The internal storage consists essentially of a single pushdown store.

2. The input formula is scanned in one direction only. Each character in the input formula is used once and only once and in sequence.

3. The algorithms translate successfully if and only if the input formula is well-formed.

Full parenthetic:	$(\sim((x_1+x_2)\cdot x_3))$
Left parenthetic:	$(\sim((x_1+x_2\cdot x_3$
Right parenthetic:	$\sim x_1+x_2)\cdot x_3))$
Parenthesis-free:	$N\ M\ x_3\ A\ x_2 x_1$

Illustration of the Various Parenthetic Notations and the Parenthesis-Free Notation

Fig. 4-3

Several limitations of both the syntax of parenthesis-free notation and the phrase structure grammar led to the development of a new model. In a natural language a well-formed subordinate qualifier, such as a phrase or clause, can be added to or taken away from a well-formed sentence with the resultant sentence remaining well-formed. This property must be reflected in a model. If a well-formed string of characters is added to or taken away from a well-formed formula in the parenthesis-free notation, the resultant formula is not well-formed. Other difficulties also arise with the phrase structure model, which was designed from the point of view of sentence synthesis rather than of sentence analysis.

To provide a theoretical basis for the analysis of natural language and to account for some of its features, a new model of natural

language, characterized by the "essential" formula (Sec. 2), which is analogous to the well-formed formula for artificial languages, is offered in this chapter. In Sec. 3 are presented several algorithms, with a Δ_M-theorem for each. Certain fundamental modifications to essential formulas are proposed in Sec. 4, and the relationship of the model to natural language is presented in Sec. 5.

The essential formula and its subsequent modification are a logical method for developing a model, corresponding in several characteristics to natural language. This model is not unique but has several attractive properties.

In the development of the algorithms (Secs. 3 and 4), Iverson's notation (Appendix A) will be used.

2. The Essential Formula

The concepts and notation of Burks, Warren, and Wright will be used wherever possible.

Consider a language characterized as follows:

Definition 1: Any finite sequence of characters, including the null sequence, is a formula.

"Λ" will designate the null formula. In general, lower case Greek letters will signify single characters, whereas upper case Greek letters will signify strings of characters or entire formulas. On occasion, formulas will be considered as vectors of characters. The following terminology will be used for formulas:

Let $\Delta = \Phi\Psi$, where the juxtaposition of "Φ" and "Ψ" denotes the concatenation of the formulas Φ and Ψ. Commas to indicate concatenated formulas may or may not be supplied.

Definition 2:

(a) The length $L(\Delta)$ of Δ is the number of characters in Δ.

(b) The head $h^i(\Delta)$ is the unique formula Φ, such that if $i \leq L(\Delta)$, then $L(\Phi) = i$; and if $i > L(\Delta)$, then $h^i(\Delta) = \Delta$.

(c) The tail $t^j(\Delta)$ is the unique formula Ψ, such that if $j \leq L(\Delta)$, then $L(\Psi) = j$; and if $j > L(\Delta)$, then $t^j(\Delta) = \Delta$.

(d) The proper head $h_p^i(\Delta)$ is the unique formula Φ, such that if $i < L(\Delta)$, then $L(\Phi) = i$.

(e) The proper tail $t_p^j(\Delta)$ is the unique formula Ψ, such that if $j < L(\Delta)$, then $L(\Psi) = j$.

A head $h(\Delta)$ or a tail $t(\Delta)$ will be written without the superscripts whenever this simpler notation is unambiguous.

Definition 3: Every character of a formula is either a functor $F_i^{(n)}$ or a variable x_i.

Definition 4: The three measures, weight (W), degree (D), and measure (M), are defined as follows:

δ	$W(\delta)$	$D(\delta)$	$M(\delta)$	
x_i	1	0	-1	
$F_i^{(n)}$	1-n	n	n	$(n > 0)$

Subscripts on a functor or a variable will be used for identification purposes and have no inherent significance. Superscripts on a functor will be used to indicate the measure of the functor.

<u>Definition 5</u>: The weight, degree, and measure of a formula are equal, respectively, to the sums of the weights, degrees, and measures of the characters of the formula.

<u>Definition 6</u>: A formula Δ is <u>essential</u> if and only if $M(\Delta) = 0$ and $M_{min}\left[h(\Delta)\right] \geq 0$.

<u>Example 1</u>. Let $\Delta_1 = F_1^{(3)}\, x_1 x_2 F_2^{(2)} x_3 x_4 x_5$.

$$M(\delta) = 3, -1, -1, 2, -1, -1, -1$$

$$M\left[h(\Delta_1)\right] = 3, 2, 1, 3, 2, 1, 0$$

Since $M(\Delta_1) = 0$ and $M_{min}\left[h(\Delta_1)\right] \geq 0$, Δ is essential. $\Delta_2 = x_1 F_1^{(3)} x_2 x_3 F_2^{(1)} x_4$ is nonessential, since $M_{min}\left[h(\Delta_2)\right] = -1$. $\Delta_3 = F_1^{(3)} x_1 F_2^{(1)} x_2 x_3$ is nonessential, since $M(\Delta_3) = 1$.

<u>Definition 7</u>: A <u>section</u> Δ_s of a formula Δ consists of any contiguous set of characters of Δ such that $L(\Delta_s) \leq L(\Delta)$. If $L(\Delta_s) < L(\Delta)$, then Δ_s is a <u>proper section</u>.

<u>Definition 8</u>: If an essential formula Δ has an essential proper section Δ_s, then Δ is <u>reducible</u>. Conversely, if Δ has no essential proper section Δ_s, Δ is <u>irreducible</u>.

<u>Example 2</u>. $\Delta_1 = F_1^{(3)} x_1 F_2^{(1)} x_2 x_3 x_4$ is an essential reducible formula with an essential proper section $F_2^{(1)} x_2$ that is irreducible.

Definition 9: Δ is a positive formula if and only if

$$W_{min}\left[t(\Delta)\right] > 0.$$

Lemma 1

Every essential formula is a positive formula.

PROOF: Consider an essential formula $\Delta = \Delta_H\Delta_T$, where $L(\Delta_T) = n > 0$. Since $M(\Delta) = 0$ and $M(\Delta_H) \geq 0$, $M(\Delta_T) \leq 0$. Consider the characters in Δ_T:

(a) if there are any functors in Δ_T,

$$-\sum_{x_i \in \Delta_T} M(x_i) \geq \sum_{F_i \in \Delta_T} M(F_i),$$

and since $M(F_i) > -W(F_i)$ by Def. 4, it follows that

$$\sum_{F_i \in \Delta_T} M(F_i) > -\sum_{F_i \in \Delta_T} W(F_i);$$

hence, since $-M(x) = W(x)$ by Def. 4,

$$\sum_{x_i \in \Delta_T} W(x_i) > -\sum_{x_i \in \Delta_T} W(F_i).$$

Therefore,

$$\sum_{x_i \in \Delta_T} W(x_i) + \sum_{F_i \in \Delta_T} W(F_i) > 0, \text{ and } W(\Delta_T) > 0.$$

(b) if there are no functors in Δ_T,

$$-\sum_{x_i \in \Delta_T} M(x_i) = \sum_{x_i \in \Delta_T} W(x_i) > 0, \text{ and } W(\Delta_T) > 0.$$

Since this holds for every n, $0 < n \leq L(\Delta)$, Δ must be positive.

Theorem 1

Every essential formula is either of the form $F^{(n)}x_n x_{n-1}, \ldots, x_2 x_1$, or else it is reducible.

PROOF: Consider an essential formula $\Delta = \Delta_H \Delta_T$, where $\Delta_T = F_r^{(s)} x_n x_{n-1}, \ldots, x_2 x_1$ and $F_r^{(s)}$ is the rightmost functor of Δ. The measure $M\left[F_r^{(s)}\right] = s$ is less than or equal to n, for otherwise Δ would not be a positive formula, as is guaranteed by Lemma 1. Therefore, $n - s + 1 \geq 1$, and there is a section, $\Delta_s = F_r^{(s)} x_n x_{n-1}, \ldots, x_{n-s+1}$, which is essential. If $s = n$ and $\Delta_H = \Lambda$, then $\Delta = \Delta_s$ and is of the form $F^{(n)} x_n x_{n-1}, \ldots, x_2 x_1$. Otherwise, Δ_s is a proper essential section of Δ (indeed, Δ_s is irreducible), and Δ is reducible.

Corollary 1

Every essential formula contains at least one functor.

Corollary 2

An essential formula with one and only one functor is irreducible.

Theorem 2a

If $\Delta = \Delta_H \Delta_s \Delta_T$ is a reducible essential formula, with a proper essential section Δ_s, then the formula $\Delta_r = \Delta_H \Delta_T$, resulting from the removal of Δ_s, is also an essential formula.

PROOF: $M(\Delta_H) \geq 0$ and $M(\Delta_s) = 0$, hence $M(\Delta_H\Delta_s) = M(\Delta_H)$. But $M(\Delta_T) = -M(\Delta_H\Delta_s)$, since $M(\Delta) = 0$. Hence, $M(\Delta_T) = -M(\Delta_H)$, and $M(\Delta_H\Delta_T) = M(\Delta_r) = 0$.

Since $M(\Delta_H) = M(\Delta_H\Delta_s)$, $M\left[h(\Delta_H)\right] \geq 0$ and $M\left[\Delta_H, \Delta_s, h(\Delta_T)\right] \geq 0$, it follows that $M_{min}\left[\Delta_H, h(\Delta_T)\right] \geq 0$ and $M_{min}\left[h(\Delta_r)\right] \geq 0$. Therefore Δ_r is an essential formula.

Example 3a. $\Delta = F_1^{(3)} x_1 F_2^{(1)} x_2 x_3 x_4$, $\Delta_s = F_2^{(1)} x_2$, and $\Delta_r = F_1^{(3)} x_1 x_3 x_4$.

Theorem 2b

If $\Delta = \Delta_H\Delta_T$ is an essential formula and Δ_s is a second essential formula, then the formula $\Delta_r = \Delta_H\Delta_s\Delta_T$, resulting from the insertion of Δ_s, is an essential formula.

PROOF: Since $M(\Delta_s) = 0$ and $M(\Delta) = 0$, $M(\Delta_r) = 0$. Since $M_{min}(\Delta_H) \geq 0$ and $M_{min}\left[h(\Delta_s)\right] \geq 0$, $M_{min}\left[\Delta_H, h(\Delta_s)\right] \geq 0$. Also, since $M_{min}\left[\Delta_H, h(\Delta_T)\right] \geq 0$, and $M(\Delta_s) = 0$, it follows that $M_{min}\left[\Delta_H, \Delta_s, h(\Delta_T)\right] = M_{min}\left[h(\Delta_r)\right] \geq 0$. Therefore Δ_r is an essential formula.

Example 3b. $\Delta = F_1^{(3)} x_1 x_2 x_3$, $\Delta_s = F_2^{(1)} x_4$, and $\Delta_r = F_1^{(3)} F_2^{(1)} x_4 x_1 x_2 x_3$.

Theorem 2 leads to the following definitions:

Definition 10: Starting with any functor in an essential formula Δ, consider as a segment $\sum(\Delta)$ the shortest section to the right of, and including, the functor, such that $M\left[\sum(\Delta)\right] = 0$.

Lemma 2

Every essential formula Δ has a segment $\sum(\Delta)$.

PROOF: An indirect proof will be used. A contradiction will be deduced from the hypothesis that an essential formula Δ does not have a segment $\sum(\Delta)$. If Δ does not have a segment $\sum(\Delta)$, then $M\left[t(\Delta)\right] > 0$, where $h^1\left[t(\Delta)\right]$ is any functor in Δ, since $M\left\{h^1\left[t(\Delta)\right]\right\} > 0$, and the variable is the only character whose measure is less than 0. If $\Delta = \left[h(\Delta),\ t(\Delta)\right]$, then $M\left[h(\Delta)\right] \geq 0$, since Δ is an essential formula. But $M(\Delta) = M\left[h(\Delta)\right] + M\left[t(\Delta)\right] > 0$, providing the contradiction.

Definition 11: Let $\Delta = \Delta_H,\ \sum(\Delta),\ \Delta_T$. If the segment $\sum(\Delta)$ is extracted from Δ, then the result of the concatenation of the residual head and tail of Δ, $P(\Delta) = \Delta_H\Delta_T$, is the residue of the original formula Δ. $\sum(\Delta)$ and $P(\Delta)$ together constitute a reduced set of the original essential formula Δ.

Lemma 3

If Δ is an essential formula, then both $\sum(\Delta)$ and $P(\Delta)$ of every reduced set are essential formulas.

PROOF: Since the first character of $\sum$ is a functor such that $M\left[h^1(\sum)\right] > 0$, and since the variable is the only type of character whose measure is less than zero, then, for the smallest group of contiguous characters to the right of and including the functor, for which $M(\sum) = 0$, it follows that $M_{min}\left[h(\sum)\right] \geq 0$ and that $\sum$ is an essential formula.

If an essential formula is divided into a segment and a residue, and the segment is an essential formula, then by Theorem 2a the residue must be an essential formula.

Definition 12: A completely reduced set of an essential formula consists of a set of irreducible essential formulas obtained by treating both the segment and the residue of a reduced set of the essential formula as essential formulas, and by iterating the process of dividing every such essential formula into a reduced set.

Definition 13: A variable is associated with a functor if the variable and functor are members of the same irreducible essential formula of a completely reduced set.

Example 4. $\Delta = F_1^{(2)}x_1\ F_2^{(3)}x_2\ F_3^{(1)}x_3x_4x_5x_6$. A reduced set of Δ is $F_2^{(3)}x_2\ F_3^{(1)}x_3x_4x_5$ and $F_1^{(2)}x_1x_6$. Another reduced set of Δ is $F_3^{(1)}x_3$ and $F_1^{(2)}x_1\ F_2^{(3)}x_2x_4x_5x_6$. A completely reduced set of Δ is $F_1^{(2)}x_1x_6$, $F_2^{(3)}x_2x_4x_5$ and $F_3^{(1)}x_3$.

Lemma 4

The completely reduced set of an essential formula containing one functor (i.e., an irreducible essential formula) is unique, namely, itself.

PROOF: Lemma 4 is an immediate consequence of Def. 12.

Lemma 5

If an essential formula Δ is divided into a reduced set consisting of a segment $\sum(\Delta)$ and of a residue $P(\Delta)$, then any irreducible essential section Δ_s of Δ must either be contained entirely within $\sum$ or lie entirely outside of $\sum$.

PROOF: Since $\sum$ and Δ each consist of contiguous characters, the only alternatives to the possibilities stated in the Lemma are that either $t_p(\Delta_s) = h(\sum)$ or that $h_p(\Delta_s) = t(\sum)$.

The former is impossible by Def. 10 and Theorem 1, since $h^1(\sum) = F_j$, and $t_p(\Delta_s)$ can contain no functor.

To prove that the latter is impossible, let $\sum = \Phi\Psi$, such that $\Psi = h_p(\Delta_s)$. $M_{min}(\Phi) \geq 0$, $M_{min}(\Psi) > 0$, and therefore $M\left[\sum(\Delta)\right] > 0$, which contradicts the definition of a segment.

Lemma 6

Let an essential formula Δ_r differ from an essential formula Δ by some irreducible essential formula Δ_s extracted from Δ_r or added to Δ by the appropriate process of Theorem 2. Consider a reduced set $\sum(\Delta_r)$ and $P(\Delta_r)$ of Δ_r:

(a) If $\sum(\Delta_r)$ contains Δ_s, and Δ is divided into a reduced set, $\sum(\Delta)$ and $P(\Delta)$, such that either $\sum(\Delta) = \Lambda$ or $h^1\left[\sum(\Delta)\right] = h^1\left[\sum(\Delta_r)\right] = F_i$, then $P(\Delta_r) = P(\Delta)$, and the residue, $P\left[\sum(\Delta_r)\right]$, of $\sum(\Delta_r)$ when Δ_s is removed, is identical to $\sum(\Delta)$.

(b) If $P(\Delta_r)$ contains Δ_s, and Δ is divided into a reduced set, $\sum(\Delta)$ and $P(\Delta)$, such that $h^1\left[\sum(\Delta)\right] = h^1\left[\sum(\Delta_r)\right] = F_i$, then $\sum(\Delta_r) = \sum(\Delta)$, and the residue, $P\left[P(\Delta_r)\right]$, of $P(\Delta_r)$ when Δ_s is removed, is identical to $P(\Delta)$.

PROOF: (of Lemma 6a)

(A) $\sum(\Delta) = \Lambda$, which is equivalent to saying that F_i is the functor of Δ_s. This is the trivial case for which $\sum(\Delta_r) = \Delta_s$, $\sum(\Delta) = \Lambda$, and $\Delta = P(\Delta) = P(\Delta_r)$.

(B) F_i is not the functor of Δ_s. $h^k\left[\sum(\Delta)\right] = h^k\left[\sum(\Delta_r)\right]$, where $\underline{h}^1/\underline{\Delta}_s = \underline{\epsilon}^{k+1}/\underline{\sum(\Delta_r)}$.

$\sum(\Delta_r) = \left[h^k\left\{\sum(\Delta)\right\}, \Delta_s, t\left\{\sum(\Delta_r)\right\}\right]$ and $\sum(\Delta) = \left[h^k\left\{\sum(\Delta)\right\}, t\left\{\sum(\Delta)\right\}\right]$.
Since $M(\Delta_s) = 0$, $M\left[h^k\left\{\sum(\Delta)\right\}, \Delta_s\right] = M\left[h^k\left\{\sum(\Delta)\right\}\right]$. Since $M\left[\sum(\Delta)\right] = M\left[\sum(\Delta_r)\right] = 0$, $M\left[t\left\{\sum(\Delta)\right\}\right] = M\left[t\left\{\sum(\Delta_r)\right\}\right]$. Also, by Theorem 2, if $\Delta = \Delta_H\Delta_T$ and $\Delta_r = \Delta_H\Delta_s\Delta_T$, then $h^1\left[t\left\{\sum(\Delta)\right\}\right] = h^1\left[t\left\{\sum(\Delta_r)\right\}\right]$. It follows that $t\left[\sum(\Delta)\right] = t\left[\sum(\Delta_r)\right]$, since both strings are identical. $P(\Delta) = P(\Delta_r)$ and $P\left[\sum(\Delta_r)\right] = \sum(\Delta)$ when $\sum\left[\sum(\Delta_r)\right] = \Delta_s$.

PROOF: (of Lemma 6b)

Since all the characters of $\sum(\Delta_r)$ are characters of Δ, $\sum(\Delta) = \sum(\Delta_r)$ by Defs. 8 and 10. $P(\Delta_r)$ differs from $P(\Delta)$ by Δ_s. Take $\sum\left[P(\Delta_r)\right]$, such that the functor of Δ_s is the first functor of $\sum\left[P(\Delta_r)\right]$; $P\left[P(\Delta_r)\right] = P(\Delta)$ by Lemma 6a, wherein the Δ, Δ_r, and Δ_s of Lemma 6a are the $P(\Delta)$, $P(\Delta_r)$, and Δ_s of Lemma 6b.

<u>Lemma 7</u>

The result of the collection of a completely reduced set of a segment $\sum$ of an essential formula Δ and of a completely reduced set of the corresponding residue P of Δ is a completely reduced set of Δ.

PROOF: The Lemma is a direct consequence of Def. 11, Def. 12 and Lemma 3.

Theorem 3

Every essential formula Δ has a unique completely reduced set.

The proof of this theorem suggests a technique for obtaining the completely reduced set of Δ.

PROOF: The proof is by induction on the number n of functors in Δ.

(a) $n = 1$: by Lemma 4.

(b) $n > 1$: assume the theorem is true for all $k < n$.

Reduce Δ into a segment $\sum(\Delta)$ and a residue $P(\Delta)$. Consider the irreducible essential segment Δ_s whose head is the rightmost functor of Δ (Theorem 1).

Case 1: The segment $\sum_1(\Delta) = \Delta_s$. By the inductive hypothesis, the residue $P_1(\Delta)$, containing n-1 functors, has a unique completely reduced set. In this case the combination of this set with Δ_s in the manner of Lemma 7 gives the desired result.

Case 2: The segment $\sum_2(\Delta) \neq \Delta_s$.

(a) $\sum_2(\Delta)$ contains Δ_s, which can be written as $\Delta_s = \sum_3\left[\sum_2(\Delta)\right]$, such that $\Delta_1 = P_2(\Delta)$ and $\Delta_2 = P_3\left[\sum_2(\Delta)\right]$. By Lemma 6a, wherein the $P_1(\Delta)$, Δ, and Δ_s of this theorem are the Δ, Δ_r, and Δ_s of Lemma 6a, $P_2(\Delta) = \Delta_1$ and is identical to the residue, $P_4\left[P_1(\Delta)\right]$, remaining when a segment, $\sum_4\left[P_1(\Delta)\right]$, starting with the same functor as $\sum_2(\Delta)$, is removed from $P_1(\Delta)$, and $\sum_4\left[P_1(\Delta)\right] = \Delta_2$.

(b) $P_2(\Delta)$ contains Δ_s, which can be written as $\Delta_s = \sum_5\left[P_2(\Delta)\right]$, such that $\Delta_1 = \sum_2(\Delta)$ and $\Delta_2 = P_5'\left[P_2(\Delta)\right]$. By Lemma 6b, wherein the $P_1(\Delta)$, Δ, and Δ_s of this theorem are the Δ, Δ_r, and Δ_s of Lemma 6b, $\sum_2(\Delta) = \Delta_1$ and is

identical to the segment, $\sum_6[P_1(\Delta)]$, starting with the same functor as $\sum_2(\Delta)$, and $P_6[P_1(\Delta)] = \Delta_2$.

By the inductive hypothesis, each of Δ_1 and Δ_2 in (a) and (b) has a unique completely reduced set. It has been shown that Δ_1 and Δ_2 are a reduced set of $P_1(\Delta)$, hence the collection of their completely reduced sets is the completely reduced set of $P_1(\Delta)$ (Lemma 7), which proves the theorem.

3. Algorithms to Test for Essential Formulas

The basic essential formula of Sec. 2 bears little resemblance to syntactic analogues in any natural language, so that additions and modifications have to be made to the initial definitions of an essential formula to bring the language model closer to natural language. The first proposed algorithm provides a mechanism for testing whether or not a formula is essential (Sec. 3A), while the next two algorithms make similar tests on modified versions of an essential formula (Sec. 3B and Sec. 3C).

A notation for paths through flow diagrams will be useful. In a flow diagram, such as Program 4-1, the expression (x,y) will be used to express any path starting at and including step x and terminating at but not including step y. If more than two symbols, for example, (x,u,v,y), are used, the path must pass through the intermediate steps, steps u and v, in order, before terminating at step y. The expression x/y indicates that there is a direct transfer from step x to step y after the operation of step x. This is shown in the diagram by an arrow.

A. The Basic Algorithm

An algorithm is introduced to test an arbitrary formula to determine whether or not it is essential. The algorithm, called Algorithm 1 (Program 4-1), provides a mechanism whereby parentheses are placed around every segment of a reduced set of the formula on the same left-to-right pass that determines whether the formula is essential.

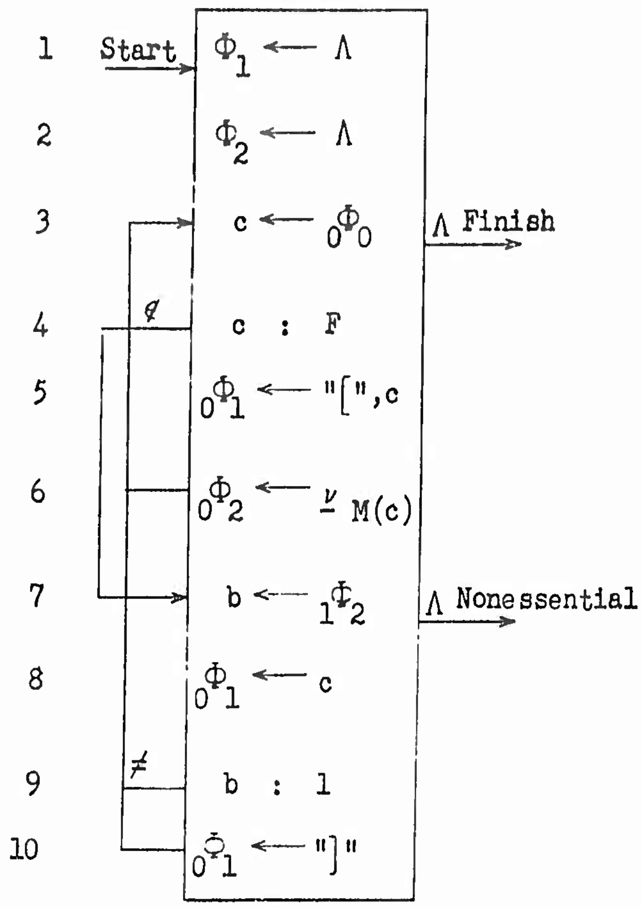
The symbols Φ_0 and Φ_1 represent the input and output files that may contain part or all of a formula (Table 4-2). Φ_1 and Φ_2 are initialized (Steps 1 and 2). In step 3, a character is read out of Φ_0. This character is identified either as a functor or a variable in step 4. If the character is a functor, a left parenthesis and the functor are written on Φ_1 (Step 5). The set of characters comprising the identity permutation vector $\underline{\nu}$, with $L(\underline{\nu}) = M(F_1)$, is written on file Φ_2 in the forward direction in step 6, after which the process returns to step 3. These steps will remain invariable, even after various restrictions are applied to essential formulas.

It should be noted that while Φ_0 and Φ_1 are read and written, respectively, in the forward direction only (corresponding to normal left-to-right reading and writing), Φ_2, the prediction pool, is written in the forward direction and read in the backward direction, that is, is written from left to right but read from right to left. The mechanism of writing in the forward direction and then reading in the backward direction is equivalent to the operation of a pushdown store. The individual characters written on Φ_2 will be referred to as <u>predictions</u>.

Φ_0	Input file containing arbitrary formula.
Φ_1	Output file.
Φ_2	Prediction pool.
Φ_3	Hindsight file (Algorithms 4 and 5 only).
c	Current character under consideration.
b,$\underline{b}$	Current prediction or set of predictions from prediction pool.
F	Set of functors.
s(x)	Class to which variable x belongs. (Algorithms 2-5 only)
$\underline{a}$	Alternative arguments of current variable. (Algorithms 4 and 5 only)
$\underline{q}$	Possible preferred arguments. (Algorithms 4 and 5 only)

Symbols for Algorithms 1 through 5

TABLE 4-2



Algorithm 1

Program 4-1

If the character being tested in step 4 is a variable, the algorithm proceeds directly to step 7 where the last character (or prediction) stored in the prediction pool is read. This prediction is one component of $\nu_{M(F_i)}$. In step 8, the character being tested is written onto the output file, and in step 9, the prediction that has just been read out of the prediction pool is tested whether or not it is the last prediction of a given set, that is, if $t^1(\Phi_2) = 1$. If so, a right parenthesis is written onto the output tape; in either case, the process returns to step 3.

Example 5. $\Delta_1 = F_1^{(3)}x_1\ F_2^{(2)}x_2\ F_3^{(2)}x_3x_4x_5x_6x_7$. After analysis, $\Delta_1 = \left[F_1^{(3)}x_1\left[F_2^{(2)}x_2\left[F_3^{(2)}x_3x_4\right]x_5\right]x_6x_7\right]$. $\Delta_2 = F_1^{(3)}x_1\ F_2^{(2)}x_2x_3\ F_3^{(2)}x_4x_5x_6x_7$. After analysis, $\Delta_2 = \left[F_1^{(3)}x_1\left[F_2^{(2)}x_2x_3\right]\left[F_3^{(2)}x_4x_5\right]x_6x_7\right]$.
$\Delta_3 = F_1^{(3)}x_1x_2x_3\ F_2^{(2)}x_4x_5\ F_3^{(2)}x_6x_7$. After analysis,
$\Delta_3 = \left[F_1^{(3)}x_1x_2x_3\right]\left[F_2^{(2)}x_4x_5\right]\left[F_3^{(2)}x_6x_7\right]$.

Several definitions referring to algorithm 1 and the succeeding algorithms are introduced:

<u>Definition 14</u>: Any path (3,3) is a <u>formula cycle</u> of Algorithm 1.

<u>Definition 15</u>: Algorithm 1 is <u>operable</u> if and only if an integral number of formula cycles are traversed. Algorithm 1 is operable for the null formula Λ.

<u>Definition 16</u>: Algorithm 1 is <u>effective</u> if an integral number of formula cycles are traversed and if $\Phi_{2\,\text{final}} = \Phi_{2\,\text{initial}}$.

<u>Definition 17</u>: Algorithm 1 is <u>strictly effective</u> if an integral number of formula cycles are traversed, and if $\Phi_{0\ \text{final}} = \Phi_{2\ \text{final}} = \Phi_{2\ \text{initial}} = \Lambda$.

The process of Algorithm 1 is continued until the terminating conditions of either step 3 or step 7 are reached. If the process terminates at step 3 and the path is strictly effective, then the formula is essential. If the algorithm is not strictly effective or if the process terminates at step 7, then the formula is nonessential (Theorem 4).

<u>Lemma 8</u>

At step 3: $L(\Phi_2) = M\left[h(\Delta)\right]$ for $M \geq 0$, where $h(\Delta)$ represents the characters of Δ that have been processed.

PROOF: In the analysis of a character of Δ, either the path (3,6/3) or the path (3,9,3) must be followed. If path (3,6/3) is followed, the character is a functor F_i, and $L(\Phi_{2\ \text{new}}) = L(\Phi_{2\ \text{old}}) + M(F_i)$. If path (3,9,3) is followed, the character is a variable x_i, and $L(\Phi_{2\ \text{new}}) = L(\Phi_{2\ \text{old}}) + M(x_i)$, where $M(x_i) = -1$. But Φ_2 is initially set to Λ, so that $L(\Phi_{2\ \text{initial}}) = 0$. Q.E.D.

<u>Lemma 9</u>

Algorithm 1 is effective for an irreducible essential segment Δ_S of a formula $\Delta = \Delta_H \Delta_S \Delta_T$ if algorithm 1 is operable for Δ_H.

PROOF: If Algorithm 1 is operable for Δ_H, step 3 is reached such that $\Phi_0 = \Delta_s\Delta_T$, and Φ_2 is an arbitrary function $g\left[\Delta_H\right]$ of Δ_H.

Let $\Delta_s = F_s^{(n)} x_n x_{n-1}, \ldots, x_2 x_1$.

After (3,6/3): $\Phi_0 = x_n x_{n-1}, \ldots, x_2 x_1 \Delta_T$, and

$$\Phi_2 = g\left[\Delta_H\right], 1,2,\ldots,n-1,n.$$

The next n-1 paths are (3,9/3). At step 9 in each formula cycle $b \neq 1$. After n-1 formula cycles, at step 3:

$$\Phi_0 = x_1\Delta_T, \text{ and}$$

$$\Phi_2 = g\left[\Delta_H\right], 1.$$

The next path is (3,10/3): $\Phi_0 = \Delta_T$, and $\Phi_2 = g\left[\Delta_H\right]$. Since $\Phi_{2\text{ final}} = \Phi_{2\text{ initial}}$, the algorithm is effective for Δ_s.

<u>Theorem 4</u> (Δ_M-theorem for Algorithm 1)

For an arbitrary $\Delta = \Delta_H\Delta_M\Delta_T \neq \Lambda$, where Δ_M is an essential formula, Algorithm 1 is effective for Δ_M if Algorithm 1 is operable for Δ_H.

PROOF: If Algorithm 1 is operable for Δ_H, step 3 is reached such that $\Phi_0 = \Delta_M\Delta_T$, and Φ_2 is an arbitrary function $g\left[\Delta_H\right]$ of Δ_H.

The proof is by induction on the number n of functors of $\Delta_M(n)$.

(a) $n = 1$: by Lemma 9;

(b) $n > 1$: assume true for all $k < n$. Consider $\sum\left[\Delta(n)\right] = \Delta_s$, an irreducible segment, and $P\left[\Delta(n)\right] = \Delta_1\Delta_2$, where $\Delta_M(n) = \Delta_1\Delta_s\Delta_2$, and $\Delta_1\Delta_2 = \Delta(n-1)$ by Lemma 7, such that $\Delta(n) = \Delta_H\Delta_1\Delta_s\Delta_2\Delta_T$.

By the inductive hypothesis on Δ_1 for $k = n-1$, step 3 is reached such that $\Phi_0 = \Delta_s\Delta_2\Delta_T$ and $\Phi_2 = g\left[\Delta_H\Delta_1\right]$.

By Lemma 9, step 3 is reached such that $\Phi_0 = \Delta_2\Delta_T$ and $\Phi_2 = g\left[\Delta_H\Delta_1\right]$.

But now, once more by the inductive hypothesis on Δ_2 for $k = n-1$, step 3 is reached such that $\Phi_0 = \Delta_T$ and $\Phi_2 = g\left[\Delta_H\right]$. Since $\Phi_{2\ final} = \Phi_{2\ initial}$, Algorithm 1 is effective for Δ_H.

Theorem 5

> Algorithm 1 is strictly effective if and only if Δ is an essential formula. A pair of parentheses is placed around every segment of a reduced set of an essential formula.

PROOF: (A) Sufficiency: by Theorem 4, if $\Delta_H = \Delta_T = \Lambda$ and $\Phi_{2\ initial} = \Lambda$.

(B) Necessity: will be shown by an indirect proof. A contradiction will be deduced from the hypothesis that the algorithm is strictly effective for a formula Δ that is not essential. Δ is not an essential formula only if either $M_{min}\left[h(\Delta)\right] < 0$ or $M(\Delta) \neq 0$.

(1) $M_{min}\left[h(\Delta)\right] < 0$. There must be a longest head Δ_1 of Δ such that $M_{min}\left[h(\Delta_1)\right] = 0$ and $M(\Delta_1) = 0$. Δ_1 can $= \Lambda$. Also, there must exist a $\Delta_2 = \Delta_1 x_i$, since the variable is the only character with a negative measure. Step 3 will be reached such that:

$$\Phi_0 = x_i, t(\Delta), \text{ and}$$

$$\Phi_2 = \Lambda$$

where $\Delta = \Delta_1, x_1$, $t(\Delta)$. The next path is (3,7), $b = \Lambda$, and the path cannot be completed.

(2) $M(\Delta) \neq 0$ and $M_{min}\left[h(\Delta)\right] \geq 0$, since otherwise the process fails by (1). Therefore, $M(\Delta) > 0$. $\Delta = h(\Delta)\ x_1x_2,\ldots,x_p$, where $t^1\left[h(\Delta)\right]$ is the rightmost functor of Δ, and $0 \leq p < M\left[h(\Delta)\right]$. Step 3 must be reached such that:

$$\Phi_0 = x_1x_2,\ldots,x_p, \text{ and}$$

$$\Phi_2 = 1,a_2,a_3,\ldots,a_q$$

where $q = M\left[h(\Delta)\right]$ by Lemma 8, hence $q > p$. After p formula cycles (3,9/3), $\Phi_2 = 1,a_2,\ldots,a_r$, where $r = q - p \neq 0$. This process will terminate at step 3 but, since $\Phi_2 \neq \Lambda$, the algorithm will not be strictly effective.

(C) Parentheses placement: Algorithm 1 is effective for any $\sum(\Delta)$ by Theorem 4, and Φ_2 need not be empty when the initial functor of $\sum(\Delta)$ is being analyzed. Since the first character of $\sum(\Delta)$ is a functor (Def. 10), a left parenthesis will be placed to the left of that functor on Φ_1 (Step 5). Since the path for the segment is effective, the last prediction read from Φ_2 is a "1". The path must end (10/3), so that a right parenthesis is placed on Φ_1 after all the characters of the segment have been written on Φ_1.

B. Ordered Variables

To make the predictions more meaningful, the variables have been restricted so that they are predicted individually and not merely counted as in Algorithm 1 (Def. 18). In natural language, the requirement that in a sentence a subject, predicate, and object occur in a given order is tantamount to the restriction of Def. 18.

Definition 18: (Def. 5 revised) A formula Δ is essential if and only if $M(\Delta) = 0$, $M_{min}\left[h(\Delta)\right] \geq 0$, and the variables associated with a functor belong to disjoint classes in the order $n, n-1, \ldots, 2, 1$ specified for a functor $F^{(n)}$.

In Sec. 4.3C the restriction will be relaxed so that the predictions need not be fulfilled in the same order as they are made.

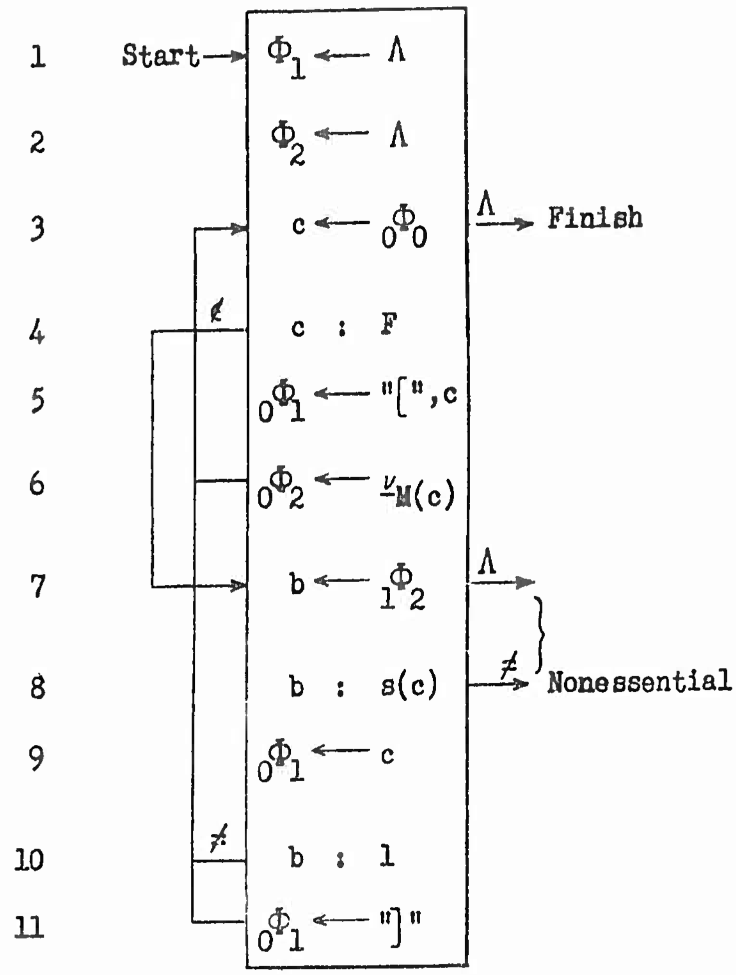
For example, consider the irreducible essential formula $\Delta = F^{(n)} x_n^n x_{n-1}^{n-1} \cdots x_2^2 x_1^1$. The class $s(x)$ to which a variable belongs is denoted by an integral superscript, x^s.

The only change to Algorithm 1 necessary to identify an ordered essential formula is the addition of step 8 of Algorithm 2 (Program 4-2), where the class to which the variable being tested belongs is compared to that of the last prediction stored in the pool (Theorems 6 and 7).

A prediction will be considered fulfilled if it is identical to the class of a variable.

Example 6. $\Delta_1 = F_1^{(3)} x_1^3 \; F_2^{(2)} x_2^2 \; F_3^{(2)} x_3^2 x_4^1 x_5^1 x_6^2 x_7^1$. After analysis, $\Delta_1 = \left[F_1^{(3)} x_1^3 \left[F_2^{(2)} x_2^2 \left[F_3^{(2)} x_3^2 x_4^1\right] x_5^1\right] x_6^2 x_7^1\right]$. However, $\Delta_2 = F_1^{(3)} x_1^3 \; F_2^{(2)} x_2^2 \; F_3^{(2)} x_3^2 x_4^2 x_5^1 x_6^1 x_7^1$ is nonessential, since $\Delta_2 = \left[F_1^{(3)} x_1^3 \left[F_2^{(2)} x_2^2 \left[F_3^{(2)} x_3^2 x_4^2 \cdots\right.\right.\right.$, and the two variables associated with F_3 belong to the same class.

The proof of Lemma 8 is valid for Algorithm 2.



Algorithm 2

Program 4-2

Lemma 10

Algorithm 2 is effective for an irreducible essential segment Δ_s of a formula $\Delta = \Delta_H\Delta_s\Delta_T$ if Algorithm 2 is operable for Δ_H.

PROOF: This proof is similar to the proof for Lemma 9. If Algorithm 2 is operable for Δ_H, step 3 is reached such that $\Phi_0 = \Delta_s\Delta_T$, and Φ_2 is an arbitrary function $g\left[\Delta_H\right]$ of Δ_H.

Let $\Delta_s = F^{(n)}x_n^n x_{n-1}^{n-1}\cdots x_2^2 x_1^1$.

After (3,6/3): $\Phi_0 = x_n^n x_{n-1}^{n-1}\cdots x_2^2 x_1^1\Delta_T$, and

$$\Phi_2 = g\left[\Delta_H\right],1,2,\ldots,n-1,n.$$

The next n-1 formula cycles are (3,10/3). At step 8 of each formula cycle, $b = s \neq 1$. After n-1 formula cycles, step 3 is reached such that:

$$\Phi_0 = x_1^1\Delta_T, \text{ and}$$

$$\Phi_2 = g\left[\Delta_H\right],1.$$

The next formula cycle is (3,11/3):

$$\Phi_0 = \Delta_T, \text{ and}$$

$$\Phi_2 = g\left[\Delta_H\right]$$

Since $\Phi_{2\ \text{initial}} = \Phi_{2\ \text{final}}$, Algorithm 2 is effective for Δ_s.

Theorem 6 (Δ_M-theorem for Algorithm 2)

For an arbitrary $\Delta = \Delta_H\Delta_M\Delta_T \neq \Lambda$, where Δ_M is an essential formula, Algorithm 2 is effective for Δ_M if Algorithm 2 is operable for Δ_H.

PROOF: The inductive proof is parallel to that of Theorem 4 (with Lemma 10 substituted for Lemma 9).

Theorem 7

Algorithm 2 is strictly effective if and only if Δ is an essential formula. A pair of parentheses is placed around every segment of a reduced set of the essential formula.

PROOF: (A) Sufficiency: by Theorem 6 if $\Delta_H = \Delta_T = \Lambda$ and $\Phi_{2\ \text{initial}} = \Lambda$.

(B) Necessity: will be shown by an indirect proof. A contradiction will be deduced from the hypothesis that the algorithm is strictly effective for a formula Δ that is not essential. Δ is not an essential formula only if:

(1) $M_{min}\left[h(\Delta)\right] < 0$, or

(2) $M(\Delta) \neq 0$, or

(3) the variables are out of order.

If conditions (1) or (2) exist, the proof is parallel to proof B in Theorem 5. If the variables are out of order, there will be a step 8 such that $b \neq s$, and the path cannot be completed.

(C) Parentheses placement: the proof is parallel to proof C of Theorem 5.

C. Relaxation of Order Restriction

If the ordering restriction (Def. 18 and Algorithm 2) on the variables is relaxed, then the top prediction in the pool need not be the only prediction which must be compared to the class of the variable being tested (Algorithm 3). For example, if $\Delta_1 = F^{(2)}x_1^2x_2^1$ and $\Delta_2 = F^{(2)}x_1^1x_2^2$,

both Δ_1 and Δ_2 can be considered essential if the ordering restriction is relaxed, while only Δ_1 would be considered essential by Algorithm 2. This is equivalent to a natural language where the subject, predicate, and object within a clause are expected to occur in a given order, but where it is possible for the order to be permuted.

<u>Definition 19</u>: (Defs. 5 and 18 revised) A formula Δ is essential if and only if $M(\Delta) = 0$, $M_{min}\left[h(\Delta)\right] \geq 0$, and the variables associated with a functor belong to disjoint classes c, where $1 \leq c \leq n$ if the functor is of measure n. The variables may occur in any order whatsoever.

In Algorithm 3 (Program 4-3) as opposed to Algorithms 1 and 2, it is necessary to search among a set of predictions in the prediction pool for fulfillment rather than merely to take the topmost prediction from the pool.

As shown by the Δ_M-theorem, it is necessary to fulfill the predictions of the rightmost analyzed functor before fulfilling the predictions of the other functors further to the left. In Algorithm 1, since the variables were merely being counted, the fulfillment of a "1" prediction in the prediction pool was an indication that the last variable associated with a given functor had been found. A right parenthesis was inserted on the output file after the variable was copied. In Algorithm 2, the indication in the prediction pool was also a "1" because of the ordering restriction. An x^1, the "last" variable associated with a given functor, could occur only after an $x^n,\cdots,x^3,x^2$ had been found for the associated functor $F^{(n)}$. After the x^1

was identified and copied, a right parenthesis could be written on the output file.

With the relaxed ordering restriction (Algorithm 3), a new device must be introduced to determine when all the variables associated with a functor have been identified. In the example cited previously, $\Delta = F^{(2)}x_1^1x_2^2$, x^1 is obviously not the "last" variable to be associated with $F^{(2)}$. A sentinel must be inserted into the prediction pool to isolate the predictions associated with different functors. All the predictions preceding the first sentinel in the prediction pool (reading in the usual right-to-left order) are tested, and any one of these can be fulfilled by a single given variable. The sentinel both restricts the variables to one member of each class that can be fulfilled, and marks the number of predictions which can be fulfilled by variables associated with a given functor, so that no more than n variables are associated with a functor $F^{(n)}$

For example, if the first two characters of $\Delta_1 = F_1^{(3)}F_2^{(2)}x_1^1x_2^3x_3^2x_4^2x_5^1$ have been analyzed, Φ_2 = s,1,2,3,s,1,2, where s represents the sentinel. Δ_1 is nonessential since x_2 belongs to class "3" and must be associated with F_2. If no sentinel were in Φ_2, the "3" prediction would be fulfilled by x_2. Likewise, if the first two characters of $\Delta_2 = F_1^{(2)}F_2^{(2)}x_1^1x_2^1x_3^2x_4^2$ have been analyzed, Φ_2 = s,1,2,s,1,2. Δ_2 is nonessential, since x_1 and x_2 both are associated with F_2 and both belong to class "1". The sentinel prevents the second "1" prediction, located to the left of the rightmost sentinel, from being fulfilled by x_2.

In Algorithm 3, the predictions generated by each functor are considered as elements of a vector associated with that functor. An _end of vector_ symbol that separates vectors written on a serial file is assumed

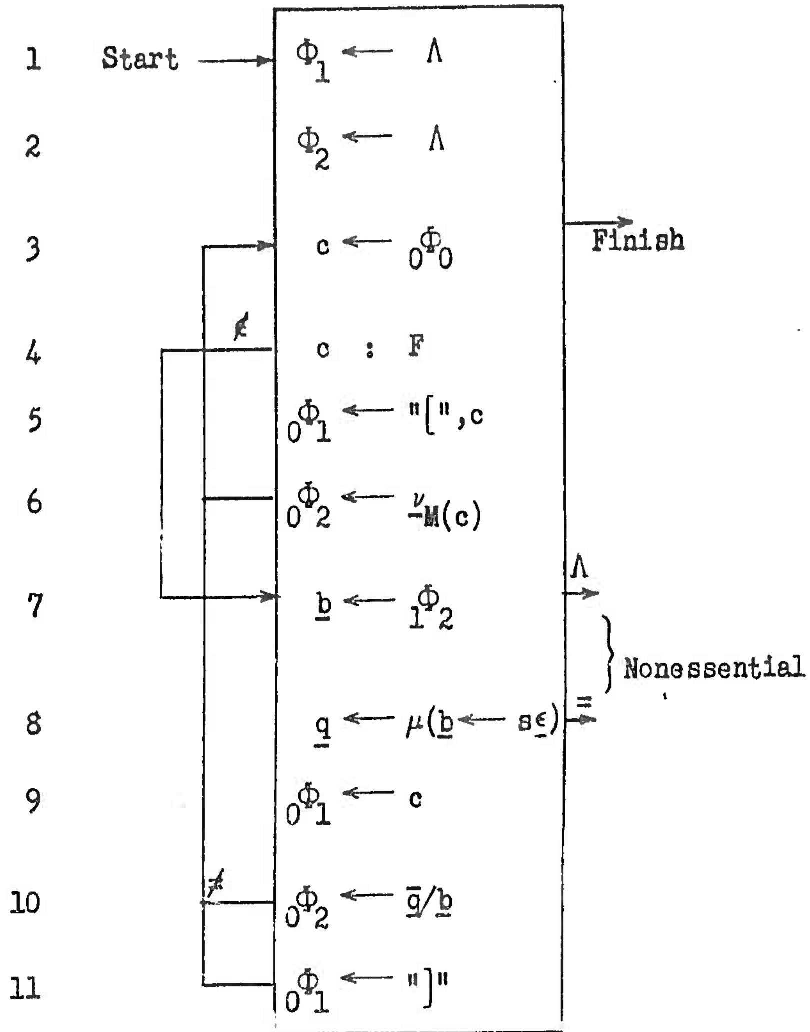
implicitly in the operation of the file. These end of vector symbols are also used as the needed sentinels in Algorithm 3.

Algorithm 2 has been modified somewhat for this purpose. Steps 1 to 6 remain unchanged. If the character under consideration is a variable, the last vector in the prediction pool is read into b in step 7. In step 8, the class to which the current variable belongs is mapped onto b, which should contain all the unfulfilled predictions associated with the rightmost not completely analyzed functor. If a prediction can be fulfilled, the variable is written on the output file in step 9, and the prediction is removed from b in step 10. If there are other predictions left in b, this is an indication that all the variables associated with the functor have been identified, so that a right parenthesis is written on the output file before the algorithm returns to step

If a variable is being tested when the prediction pool is empty, the formula is nonessential. If the prediction of a variable being tested cannot be found in the prediction pool (step 8), the formula is also nonessential (Theorem 9).

Example . $\Delta_1 = F_1^{(3)}x_1^2\ F_2^{(2)}x_2^2\ F_3^{(2)}x_3^1x_4^2x_5^1x_6^1x_7^3$. After analysis, $\Delta_1 = \left[F_1^{(3)}x_1^2\left[F_2^{(2)}x_2^2\left[F_3^{(2)}x_3^1x_4^2\right]x_5^1\right]x_6^1x_7^3\right]$. $\Delta_2 = F_1^{(3)}x_1^1\ F_2^{(2)}x_2^2\ F_3^{(2)}x_3^2x_4^1x_5^1x_6^2x_7^3$. After analysis, $\Delta_2 = \left[F_1^{(3)}x_1^1\left[F_2^{(2)}x_2^2\left[F_3^{(2)}x_3^2x_4^1\right]x_5^1\right]x_6^2x_7^3\right]$.

$\Delta_3 = F_1^{(3)}x_1^3\ F_2^{(2)}x_2^2\ F_3^{(2)}x_3^1x_4^1x_5^2x_6^1x_7^3$. Δ_3 is nonessential, since $\Delta_3 = \left[F_1^{(3)}x_1^3\left[F_2^{(2)}x_2^2\left[F_3^{(2)}x_3^1x_4^1\right.\right.\right.\cdots$ and the two variables associated with F_3 belong to the same class. $\Delta_4 = F_1^{(3)}x_1^2\ F_2^{(2)}x_2^3x_3^2x_4^1x_5^1$. Δ_4 is nonessential, since



Algorithm 3

Program 4-3

$\Delta_4 = \left[F_1^{(3)} x_1^2 \left[F_2^{(2)} x_2^3 \cdots\right.\right.$ and a vector belonging to class "3" is associated with a functor $F^{(2)}$.

The proof of Lemma 8 is valid for Theorem 3.

Lemma 11

Algorithm 3 is effective for an irreducible essential segment Δ_s of a formula $\Delta = \Delta_H \Delta_s \Delta_T$ if Algorithm 3 is operable for Δ_H.

PROOF: The proof is similar to the proofs of Lemmas 9 and 10. If Algorithm 3 is operable for Δ_H, step 3 is reached such that $\Phi_0 = \Delta_s \Delta_T$ and Φ_2 is an arbitrary function $g\left[\Delta_H\right]$ of Δ_H.

Let $\Delta_s = F^{(n)} x_n^{s_n} x_{n-1}^{s_{n-1}} \cdots x_2^{s_2} x_1^{s_1}$, where $s_i \neq s_j$ for all $i \neq j$, $0 < s_i \leq n$, and $0 < i \leq n$.

After (3,6/3): $\Phi_0 = x_n^{s_n} x_{n-1}^{s_{n-1}} \cdots x_2^{s_2} x_1^{s_1} \Delta_T$, and

$$\Phi_2 = \left\{g\left[\Delta_H\right]\right\}\left\{1,2,\ldots,n-1,n\right\}.$$

The next n-1 formula cycles are (3,10/3). At step 8, in each formula cycle, there is a $b_i = s$ where $i \leq L(\underline{b})$; also $\overline{\underline{q}}/\underline{b} \neq \Lambda$. After n-1 formula cycles, step 3 is reached such that:

$$\Phi_0 = x_1^{s_1} \Delta_T, \text{ and}$$

$$\Phi_2 = \left\{g\left[\Delta_H\right]\right\}, \left\{s_1\right\}.$$

The next formula cycle is (3,11/3):

$$\Phi_0 = \Delta_T, \text{ and}$$

$$\Phi_2 = \{g[\Delta_H]\}.$$

Since $\Phi_{2\ \text{initial}} = \Phi_{2\ \text{final}}$, Algorithm 3 is effective for Δ_s.

<u>Theorem 8</u> (Δ_M-theorem for Algorithm 3)

For an arbitrary $\Delta = \Delta_H\Delta_M\Delta_T \neq \Lambda$, where Δ_M is an essential formula, Algorithm 3 is effective for Δ_M if Algorithm 3 is operable for Δ_H.

PROOF: The inductive proof is parallel to that of Theorem 4 (with Lemma 11 substituted for Lemma 9).

<u>Theorem 9</u>

Algorithm 3 is strictly effective if and only if Δ is an essential formula. A pair of parentheses is placed around every segment of a reduced set of an essential formula.

PROOF: (A) Sufficiency: by Theorem 8, if $\Delta_H = \Delta_T = \Lambda$ and $\Phi_{2\ \text{initial}} = \Lambda$.

(B) Necessity: will be shown by an indirect proof. A contradiction will be deduced from the hypothesis that the algorithm is strictly effective for a formula Δ that is not essential.

Δ is not an essential formula only if either:

(1) $M_{min}[h(\Delta)] > 0$, or

(2) $M(\Delta) \neq 0$, or

(3) there are two or more variables belonging to the same class associated with one functor, or

(4) there is a variable belonging to class "n" associated with a functor $F^{(m)}$, where $m < n$.

If conditions (1) or (2) exist, the proof is parallel to proof B of Theorem 5. If condition (3) exists, there are an x_1^j and an x_2^j (x_1 preceding x_2) associated with one functor F_n, such that when $c = x_1^j$ after step 10: $b_i \neq j$ for any b_i left in $\underline{b}$. When $c = x_2^j$, at step 8: $\underline{q} = 0$ and the path cannot be completed. If condition (4) exists, when x^n is being tested, there will be no "n" in $\underline{b}$ and $\underline{q} = 0$, so that the path cannot be completed.

(C) Parentheses placement: the proof is parallel to proof C of Theorem 5.

4. Further Modifications to the Essential Formula

It has been assumed in the model as developed in Sec. 3 that every variable is a member of only one class, so that when a variable is being tested only this one class is tested against the predictions in the pool. In this section, the problem of a variable belonging to more than one class will be considered. This is analogous in natural language to the possibility of a word having more than one role. For example, in English, the word "water" might refer, on the one hand, to the liquid, in which case "water" is a noun or on the other hand, to the act of feeding plants, in which case "water" is a verb.

The outcome of the modifications to be set forth in this section is that a single pass[†] through a formula will not necessarily be sufficient to determine whether or not the formula is essential. On occasion it will be necessary to make several passes before this is determined. Algorithms, extended from those of the last section, will be given for a single pass of the formulas being tested. Analogues of the theorems of Algorithms 1 to 3 do not exist for a single pass of Algorithm 4. The development of an algorithm that will control the iteration of a sentence is a fruitful field for further research. Meaningful theorems should be obtainable from such a study.

A. Multi-class Variables

In Algorithm 3, if each variable can belong to only one class, and if a prediction of that class is in a location in the prediction pool where it can be fulfilled, the variable being tested is accepted, and the algorithm proceeds to test the following character. If there is no appropriate prediction that can be fulfilled, the variable is not accepted and the entire formula is rejected as nonessential.

To take into account the possibility of a variable belonging to more than one class, the following definitions will prove to be helpful.

[†] The analysis of a formula, which tests each character in the order of occurrence once and only once, is defined as a <u>pass</u>. The set of passes required to determine whether or not a formula is essential is defined as an <u>iteration</u>.

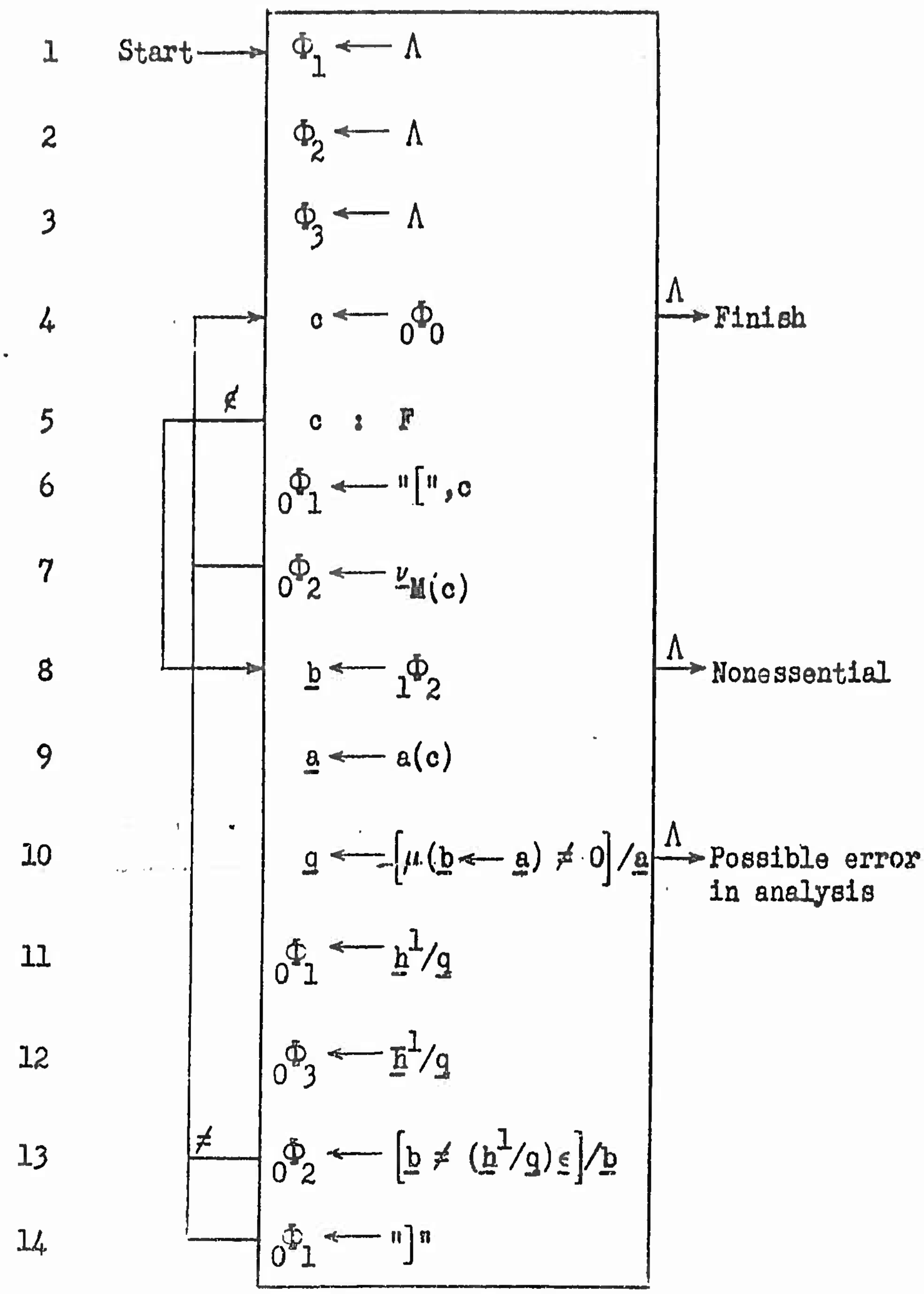
<u>Definition 20</u>: A variable $x^{\alpha,\beta,\gamma}$ can belong to any of the classes α, β, and γ, where each of the classes is an <u>argument</u> of x and each member of a set of classes is an <u>alternative argument</u> of x.

<u>Definition 21</u>: The class to which a variable with alternative arguments is assigned in the process of a syntactic analysis of an essential formula is the <u>preferred argument</u>.

Whereas alternative arguments of a variable are known qualities of the variable being tested by the algorithm, the preferred argument is selected from the alternative arguments according to the contents of the prediction pool at the time of the test (Algorithm 4).

If it is assumed that there is no <u>a priori</u> preference for any alternative argument or for any prediction in the pool, then all the alternative arguments are compared with all the predictions preceding the first sentinel (steps 8-10). When all the possible preferred arguments are found, one of them is selected arbitrarily and entered on the output file as the preferred argument (step 11). All others are recorded onto a <u>hindsight</u> or temporary storage file (step 12). The prediction that was fulfilled by the preferred argument is then removed from the prediction pool (step 13), and this process is continued with the next character. When all the variables associated with a given functor have been identified, a right parenthesis is written on the output file (step 14).

This process must end with one of the three terminal conditions of the algorithm. If the algorithm is strictly effective, then the algorithm



Algorithm 4

Program 4-4

has successfully identified the formula as an essential formula and has selected a preferred argument for each variable. If the process ends at step 10, then the particular preferred arguments that were chosen did not lead to the evaluation of an essential formula, but some other selection of a preferred argument might possibly lead to the desired evaluation. If the process ends either at step 4 and the path is not strictly effective or at step 8, the formula is definitely nonessential.

If there is a choice of alternative arguments at step 11, it is impossible to determine whether the appropriate one is chosen as the preferred argument. Therefore, even if a strictly effective evaluation was chosen, other alternative evaluations must be tried, since there might be one or even more than one additional evaluation for which the algorithm is strictly effective. Information about the alternative paths is available on the hindsight file, since every time a branching point in the analysis occurs, all the alternative preferred arguments, except for the one selected, are recorded there.

Example 8. $\Delta_1 = F^{(3)}x_1^3 x_2^{1,2} x_3^2$. There are two possible analyses of Δ_1. Either $\Delta_1^{(1)} = \left[F^{(3)}x_1^3 x_2^1 x_3^2\right]$ or $\Delta_1^{(2)} = \left[F^{(3)}x_1^3 x_2^2 x_3^2\right]$. $\Delta_1^{(1)}$ is an essential formula but $\Delta_1^{(2)}$ is not. Since there are no other alternative evaluations, a unique argument can be assigned to each variable.

$\Delta_2 = F^{(3)}x_1^{3,2} x_2^{3,2} x_3^1$. There are two analyses that lead to an essential formula: both $\Delta_2^{(1)} = \left[F^{(3)}x_1^3 x_2^2 x_3^1\right]$ and $\Delta_2^{(2)} = \left[F^{(3)}x_1^2 x_2^3 x_3^1\right]$. Therefore, a unique argument cannot be assigned to each variable. $\Delta_3 = F^{(3)}x_1^{3,2} x_2^{3,2} x_3^2$. Δ_3 is nonessential, since no matter what evaluation is undertaken, an essential formula cannot be found.

If an algorithm to keep track of alternative paths were available and Algorithm 4 could be applied iteratively until either all the possible combinations of preferred arguments were tried or until a terminal were reached indicating that the formula was definitely nonessential, either none, one, or more than one of these combinations would lead to a satisfactory interpretation. If none of the combinations resulted in an essential formula, then the formula would be nonessential. If one and only one combination resulted in an essential formula, the formula would be essential and unique preferred arguments would have been assigned to each variable. If more than one combination resulted in an essential formulation, the formula would be essential but not all the variables could be assigned unique preferred arguments.

B. All Predictions Need Not be Fulfilled

It has previously been assumed that all the predictions in the pool are fulfilled if a formula is essential. However, in natural language, if, say, an object prediction is made for every clause, a clause without an object should not be rejected.

It is now assumed that, although $M(F_1)$ is known, there need not be as many as M variables associated with the functor. When the alternative arguments of a variable do not correspond to any predictions remaining in the pool preceding the first sentinel, but do correspond to a prediction following the first sentinel, this is now the only indication that all the variables associated with a functor have been identified (Algorithm 5).

1	Start →	$\Phi_1 \leftarrow \Lambda$	
2		$\Phi_2 \leftarrow \Lambda$	
3		$\Phi_3 \leftarrow \Lambda$	
4		$c \leftarrow {}_0\Phi_0$	Λ → Finish
5	$\neq$	$c : F$	
6		${}_0\Phi_1 \leftarrow$ "[", c	
7		${}_0\Phi_2 \leftarrow \overset{\nu}{\underline{M}}(c)$	
8		$\underline{b} \leftarrow {}_1\Phi_2$	Λ → Possible error in analysis
9		$\underline{a} \leftarrow a(c)$	
10		$\underline{g} \leftarrow [\mu(\underline{b} \leftarrow \underline{a}) \neq 0]/\underline{a}$	$\neq$
11		${}_0\Phi_1 \leftarrow$ "]"	
12		${}_0\Phi_1 \leftarrow \underline{h}^1/\underline{g}$	
13		${}_0\Phi_3 \leftarrow \bar{\underline{h}}^1/\underline{g}$	
14	$\neq$	${}_0\Phi_2 \leftarrow [\underline{b} \neq (\underline{h}^1/\underline{g})\underline{\epsilon}]/\underline{b}$	
15		${}_0\Phi_1 \leftarrow$ "]"	

Algorithm 5

Program 4-5

The most striking difference between Algorithm 4 and Algorithm 5 is that there is one less terminal condition in the latter algorithm. That a formula is nonessential can no longer be determined on a single pass. If the algorithm ends at step 4 and is not strictly effective, or if the algorithm terminates at step 8, it is merely an indication that the chosen combination of preferred arguments is not an essential formula.

Whereas in Algorithm 4, if $\underline{q} = \Lambda$ (step 10), there was an indication that the chosen evaluation did not lead to an essential interpretation, in Algorithm 5 it is necessary to assume that all the variables associated with a given functor have been identified, to write a right parenthesis on the output file, and to bring in the next set of predictions from the prediction pool. The two algorithms are otherwise identical.

<u>Example 9</u>. $\Delta_1 = F_1^{(2)} x_1^2 \, F_2^{(3)} x_2^{3,1} x_3^1 x_4^1$. After analysis, there is only one essential formulation of Δ_1: $\Delta_1 = \left[F_1^{(2)} x_1^2 \left[F_2^{(3)} x_2^3 x_3^1\right] x_4^1\right]$, and there is no x^2 associated with F_2.

$\Delta_2 = F_1^{(2)} x_1^2 \, F_2^{(3)} x_2^3 x_3^1 x_4^{1;2}$. There are two essential formulations of Δ_2. Either $\Delta_2^{(1)} = \left[F_1^{(2)} x_1^2 \left[F_2^{(3)} x_2^3 x_3^1 x_4^2\right]\right]$ with no x^1 associated with F_1, or $\Delta_2^{(2)} = \left[F_1^{(2)} x_1^2 \left[F_2^{(3)} x_2^3 x_3^1\right] x_4^1\right]$ with no x^2 associated with F_2.

C. Prediction Span Indicator

A <u>prediction span indicator</u>, a device not used in any of the algorithms, can be assigned to each type of prediction to indicate whether or not an algorithm is leading to a nonessential solution.

An analogous situation in natural language is that of the prediction of a genitive modifier by a noun. Since the modifier need not

occur, the prediction must be marked so that if it remains unfulfilled, the analysis is not judged incorrect.

This index specifies whether a given prediction may remain unfulfilled in the analysis of an essential formula. When a given variable is being tested, it is possible that a preferred argument cannot be selected on the basis of the predictions preceding the first sentinel in the prediction pool. In this case, before the unfulfilled predictions and the sentinel are wiped from the pool (transfer(11/8)in Program 4-5), so that the algorithm can continue to test on the next set of predictions in the pool, the prediction span indicators of the unfulfilled predictions are tested. If any of the wiped predictions require fulfillment, this is sufficient indication that the selected preferred arguments are not leading to an essential formula.

5. Correlation of the Essential Formula Model with Natural Language

To correlate the model with natural language, the structure and analysis of natural language will be put into abstract algebraic terms. A sentence in a natural language consists of a finite set of elements in a given order. Since, in general, a word tested out of context gives no information about the neighboring words, the elements of the sentence may be considered as variables. A sentence can then be described as a sequence, $S = \left[x_1, x_2, \ldots, x_n\right]$, where words, punctuation marks, as well as other symbols are its elements.

The set of alternative arguments associated with each word can be retrieved from a dictionary (such as the Russian-English automatic dictionary described in Chap. 3). The information available for syntactic analysis can

be expressed as follows:

$$S = \left[x_1^{\alpha_1,\beta_1,\gamma_1,\ldots}, x_2^{\alpha_2,\beta_2,\gamma_2,\ldots}, \ldots, x_n^{\alpha_n,\beta_n,\gamma_n,\ldots}\right],$$

where the right superscripts represent the alternative arguments of x_i. It should be noted that functors are not explicit in this representation of a sentence.

The method of predictive syntactic analysis consists of the selection of a preferred argument from the predictions in the pool. The arrival at a syntactic analysis of a sentence, including the establishment of relationships among the words in the sentence, implies that the "functors" are recognized in the analysis. The functors cannot be determined by an examination of the individual words; their occurrence can only be established from the preferred argument and the prediction which selected it.

If q_{ij} represents the preferred argument of word x_j selected by prediction p_i from among the alternative arguments $\alpha_j,\beta_j,\gamma_j,\ldots$, then the relationship of the functor to the preferred argument and to the prediction can be formalized as $F_j = F_j(p_i,q_{ij})$, where F_j, as a function of p_i and q_{ij}, represents the role played by x_j in its environment in a particular sentence. An analyzed sentence S_A can then be represented by

$$S_A = \left[{}^{p_{k_1}}x_1^{q_{k_1,1}}, {}^{p_{k_2}}x_2^{q_{k_2,2}}, \ldots, {}^{p_{k_n}}x_n^{q_{k_n,n}}\right],$$

$k_i < i$, where k_i is the index of the variable making the prediction that has selected the preferred argument. The preferred argument is denoted by

the right superscript, and the prediction selecting the preferred argument is denoted by the left superscript.

Each word in a sentence now has two functions: (1) it assumes the role of a variable in fulfilling a prediction previously placed in the prediction pool; and (2) as a function of p_i and q_{ij}, it assumes the role of a functor, and makes further predictions which will be placed in the prediction pool.

The representation of an analyzed sentence is an attempt to illustrate what is known about the sentence and its individual words as it is being analyzed. Obviously, some information is known about a sentence before the analysis of the sentence even begins. For example, every sentence is expected to have a subject and a predicate as well as a period or some other punctuation mark denoting its completion. An <u>initial</u> symbol is introduced to denote this information so that

$$S_A = \left[I, {}^{p_{k_1}}_{\;1}x_1^{q_{k_1},1}, {}^{p_{k_2}}_{\;2}x_2^{q_{k_2},2}, \ldots, {}^{p_{k_n}}_{\;n}x_n^{q_{k_n},n} \right].$$

To complete the correlation of the essential formula model with this notion of a natural language, a linkage or merger of every functor with the immediately preceding variable is hypothesized. The variable then becomes the representation for a word, and the functor becomes part of this representation and need not be considered a separate entity. In Example 10, the merger of a functor and the variable immediately preceding it is indicated by a pair of slurs, $\frown$.

<u>Example 10</u>. $\Delta = \Lambda F_1^{(2)} x_1^1 \, F_2^{(3)} x_2^1 x_3^3 x_4^2 \, F_3^{(1)} x_5^1 x_6^2$. After analysis,

$$\Delta = \Lambda\left[F_1^{(2)} \; x_1^1 \left[F_2^{(3)} \; x_2^1 x_3^3 x_4^2\right]\left[F_3^{(1)} \; x_5^1\right] x_6^2\right].$$

The sentence represented by this example would be $\left[x_1^1, x_2^1, x_3^3, x_4^2, x_5^1, x_6^2\right]$, where x_1^1 and x_6^2 are selected by the initial predictions; x_2^1, x_3^3, and x_4^2 are selected by predictions generated by x_1; and x_5^1 is selected by a prediction generated by x_4.

6. Conclusions

Although the formal development of the model stems from several previously published papers, the main inspiration came from a careful study of Rhodes' empirical predictive syntactic analysis technique, as applied to Russian.

It is assumed that the structure of the Russian language is <u>nested</u> in the manner of the Δ_M-theorem. That is, if a sentence is interrupted by a phrase or clause, the embedded phrase or clause will have been analyzed completely before the analysis returns to the main part of the sentence. The phrase or clause will have no effect on the words following it. This nesting feature was brought out in the theorems beginning with Theorem 2, where it was shown that an essential segment, a nested structure, could be removed from an essential formula, leaving the resulting formula essential. The unique decomposition theorem (Theorem 3) indicated that a sentence in the model, like most sentences in the Russian language, could be decomposed uniquely into its phrases and clauses.

In the experimental program (Chap. 5), it will prove convenient to extend the concept of nesting in natural language. Individual phrases and clauses can be considered as structures within which nesting can occur. For example, a clause can be divided into three nested structures: all the

words constituting the subject, the predicate, and the object. Each of these structures might contain other nested structures. Therefore, if the sentence is to remain grammatically complete, a random nested structure cannot be removed from it.

Theorems 2 and 3 also point out the main difference between a well-formed formula and an essential formula. Any well-formed segment of a well-formed formula is firmly connected to the larger structure of the formula. The well-formed segment can be removed and a simple variable substituted in its place, but some symbol must remain to indicate the presence of the well-formed segment in the original formula. At the same time an essential section in an essential formula represents a structure completely subordinate to a variable, which in turn is tied to the larger structure of the formula. Whether or not the subordinate structure is present is immaterial.

This difference can be best illustrated with two examples. Consider the well-formed formula $\Delta_1 = \left[F_1^{(2)}x_1\left[F_2^{(2)}x_2x_3\right]\right]$, where the parentheses are used to indicate the well-formed segments in the formula. (The individual variables are well-formed formulas, but their parentheses have been omitted for clarity.) The well-formed segment $F_2^{(2)}x_2x_3$ can be replaced by a variable, but the complete absence of the segment with no substitute would render Δ_1 non-well-formed. In contrast, consider the essential formula $\Delta_2 = \left[F_1^{(2)}x_1\left[F_2^{(2)}x_2x_3\right]x_4\right]$. The essential segment $F_2^{(2)}x_2x_3$ can be removed from the formula and the variables x_1 and x_4 will remain to satisfy the predictions from F_1, and $\left[F_1^{(2)}x_1x_4\right]$ still will be an essential formula.

A second assumption made about the Russian language was that the syntactic role of a nested structure in the larger framework in which it is embedded could be completely determined by the syntactic role played by its first word. Exceptions to this assumption exist in the language, but it seems that they occur rarely enough to permit their analysis by more circuitous methods without a sacrifice of the efficiency of the predictive syntactic method as a whole. In the model this first word of a nested structure is represented by the variable which also takes on the role of a functor. As a variable, it fulfills the role of the entire nested structure in the larger structure. As a functor, the first word forms the ties to bring together all the words within the nested structure.

Such an assumption cannot be made consistently about the English language. When the two Russian noun phrases, большой дом and большие дома, are compared with their English counterparts, "the big house" and "the big houses", it can be seen that, in the Russian phrases, number (singular and plural, respectively) is indicated by the paradigmatic forms of the adjectives, whereas number is not indicated by the adjectives in the English phrases. Also, the paradigmatic forms of the Russian adjectives indicate case, information that is completely lacking in the English equivalents. To determine the complete specifications of the English noun phrases, it is necessary to look at the nouns as well as at the adjectives preceding them.

A partial verification of the usefulness of the model of the essential formula will be presented in the experimental results described in Chap. 5.

REFERENCES

1. Rhodes, I., "A New Approach to the Mechanical Syntactic Analysis of Russian," Unpublished Report, National Bureau of Standards (1959).

2. Burks, A. W., Warren, D. W., and Wright, J. B., "An Analysis of a Logical Machine Using Parenthesis-free Notation," Mathematical Tables and Other Aids to Computation, Vol. 8 (1954), pp. 53-7.

3. Chomsky, N., "Three Models for the Description of Language," IRE Transactions on Information Theory, Vol. IT-2 (1956), pp. 113-24.

4. Chomsky, N., Syntactic Structures, Mouton and Co., The Hague (1957).

5. Oettinger, A. G., "Automatic Syntactic Analysis and the Pushdown Store," Symposium on the Structure of Language and its Mathematical Aspects, 567th Meeting of the American Mathematical Society, New York (April 1960) (to appear in Proceedings of the Symposium, American Mathematical Society, Providence, Rhode Island).

6. Samelson, K. and Bauer, F. L., "Sequential Formula Translation," Communications of the Association for Computing Machinery, Vol. 3, No. 2 (1960), pp. 76-83.

CHAPTER 5

AN EXPERIMENTAL SYNTACTIC ANALYZER

1. Introduction

The experimental syntactic analyzer presented in this chapter is a system[†] that syntactically analyzes Russian sentences by a left-to-right pass utilizing the predictive syntactic analysis technique discussed in the preceding chapter. The present experimental program, which was written in January 1960, will be discussed from the point of view of several problem areas. The discussion of these areas should provide an adequate indication of the approach of predictive analysis, as well as the more pertinent details of operation, but no systematic attempt will be made to consider all the aspects of the program in complete detail.

The various rules by which this program operates constitute a verifiable although incomplete grammar of the Russian language. Traditional grammars abound with exceptions to the rules that are stated. The grammatical rules that are used in the syntactic analyzer will have to account for these exceptions if all sentences are to be analyzed by the program. Thus, it is necessary to find broad rules which govern the behavior of the exceptions as well as the more usual occurrences. Through these rules, the main goal of the experimental analyzer is to eliminate any ambiguity in the syntactic roles that are played by the words in a sentence. As the program is improved, the grammar of the program will better approximate the grammar of the Russian language.

[†]The program for this system was written by W. Bossert[1].

The experimental program is not a method for obtaining rules, as is the proposed trial translator or algorithm finder of Giuliano.[2] The only limitations on rules to be utilized in predictive analysis are that the words in the sentence under analysis must be scanned in a left-to-right order, and that the predictions must be stored in such a manner as to adhere to the basic nesting characteristic (Chapter 4) which, according to Yngve's hypothesis,[3] is applicable to many natural languages. Within these constraints, anything can be tried. A continuous attempt is made to keep the rules as systematic as possible in order to keep the data handling mechanism to a minimum. The rules that have been adopted to date in the experimental program are due to a knowledge of the Russian language systematically organized in existing grammars, elicited from native informants, and obtained as a consequence of earlier experiments.

After new rules are developed, the experimental program existing at that time is modified so that the new rules are incorporated. Several texts are analyzed with the revised program, and the output is then studied to determine whether the theories expressed by the new rules have been substantiated. There are usually many exceptions to new rules. These exceptions become obvious when the new rules are applied systematically to several texts, and then newer, more complete rules can be established.

Many of the subroutines used in the experimental program are named after classical grammatical terms, such as <u>subject</u> prediction. All of these classifications are explicitly defined within the context of the experimental program. These definitions need not coincide with the classical grammatical definitions, but they resemble the classical definitions closely.

Although it is not necessary to analyze many texts to determine the faults and limitations of a version of the experimental program, it is dangerous to reprocess the same texts for more than a few versions of the program. If the same texts are used repeatedly, the syntactic analysis program becomes a program specifically designed to analyze the writing styles of the several authors of the test texts. All the illustrations of actual analyzed output have been taken from text 00A[4], a text that was used for two versions of the experimental program and is therefore not suitable as test material for future versions.

In the discussion of this chapter it is essential to distinguish errors from mistakes. An <u>error</u> is a faulty decision in the experimental program which leads to an incorrect analysis of a sentence where the difficulty is recognized by some technique in the program. A <u>mistake</u> is a similar faulty decision where there is no indication that an incorrect analysis has been made.

In this chapter, the mechanism of predictive analysis is introduced with the analysis of two short sentences by a greatly simplified version of the present program (Sections 2 and 3). The details of the experimental program are presented in Section 4. The following four sections (Sections 5 through 8) are devoted to discussions of examples of output that demonstrate various interesting features of the program; and a brief summary of problems that are still to be solved is given in Section 9.

2. An Illustration of Predictive Syntactic Analysis

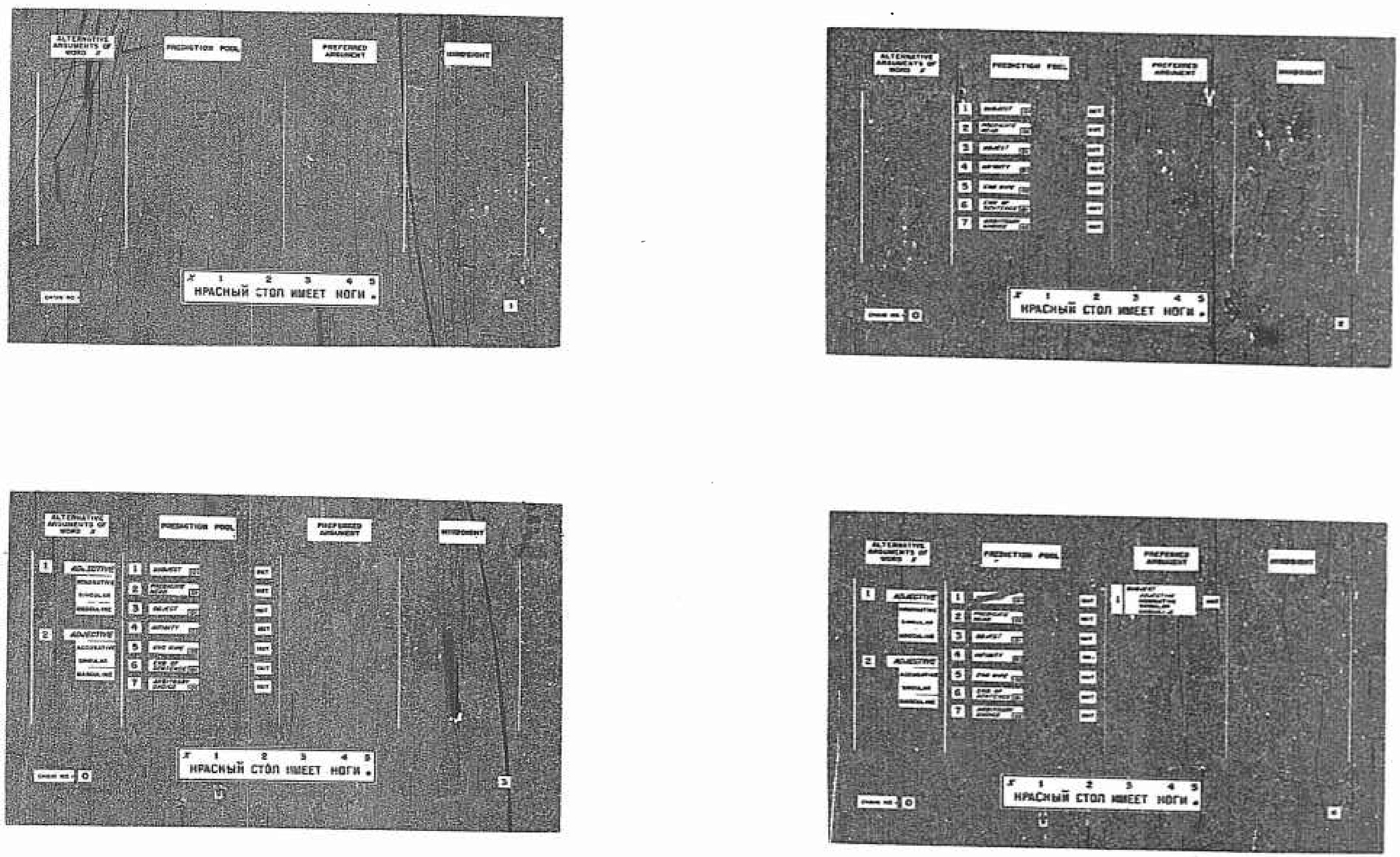
The method of predictive syntactic analysis will be exemplified by the analysis of the simple sentence: Красный стол имеет ноги. (Fig. 5-1).

To make the analysis procedure more lucid, a greatly simplified version of the analysis technique is illustrated. The number of predictions in the prediction pool is reduced, and only a small but essential fraction of the predictions is depicted. The experimental system will be discussed in Sections 4 to 9.

The format of Fig. 5-1 is indicative of the information that is stored in the computer memory, although, obviously, in the memory the information need not be literall spelled out. Seven concepts introduced in Chapter 4 have been utilized in this representation.

(1) Alternative argument - The starting point of predictive analysis is the information about the arguments of words that is obtainable from a dictionary. Since the lexical properties of words do not always define a unique argument, a set of alternative arguments must be considered. An alternative argument will be noted in this chapter by a pair of slashes; thus, столы has two alternative arguments, /noun, nominative, plural, masculine/ and /noun, accusative, plural, masculine/. This concept of argument and alternative argument is completely parallel to Definition 20 of Section 4.4A.

(2) Prediction pool - The program analyzes every word in a sentence by attempting to fulfill predictions which are potential grammatical relationships among the words of a sentence. The predictions are stored in a prediction pool which is operated approximately as a pushdown store, in the sense that the last prediction entered into the pool is the first one tested for fulfillment.



A Simplified Example of Predictive Syntactic Analysis

Fig. 5-1

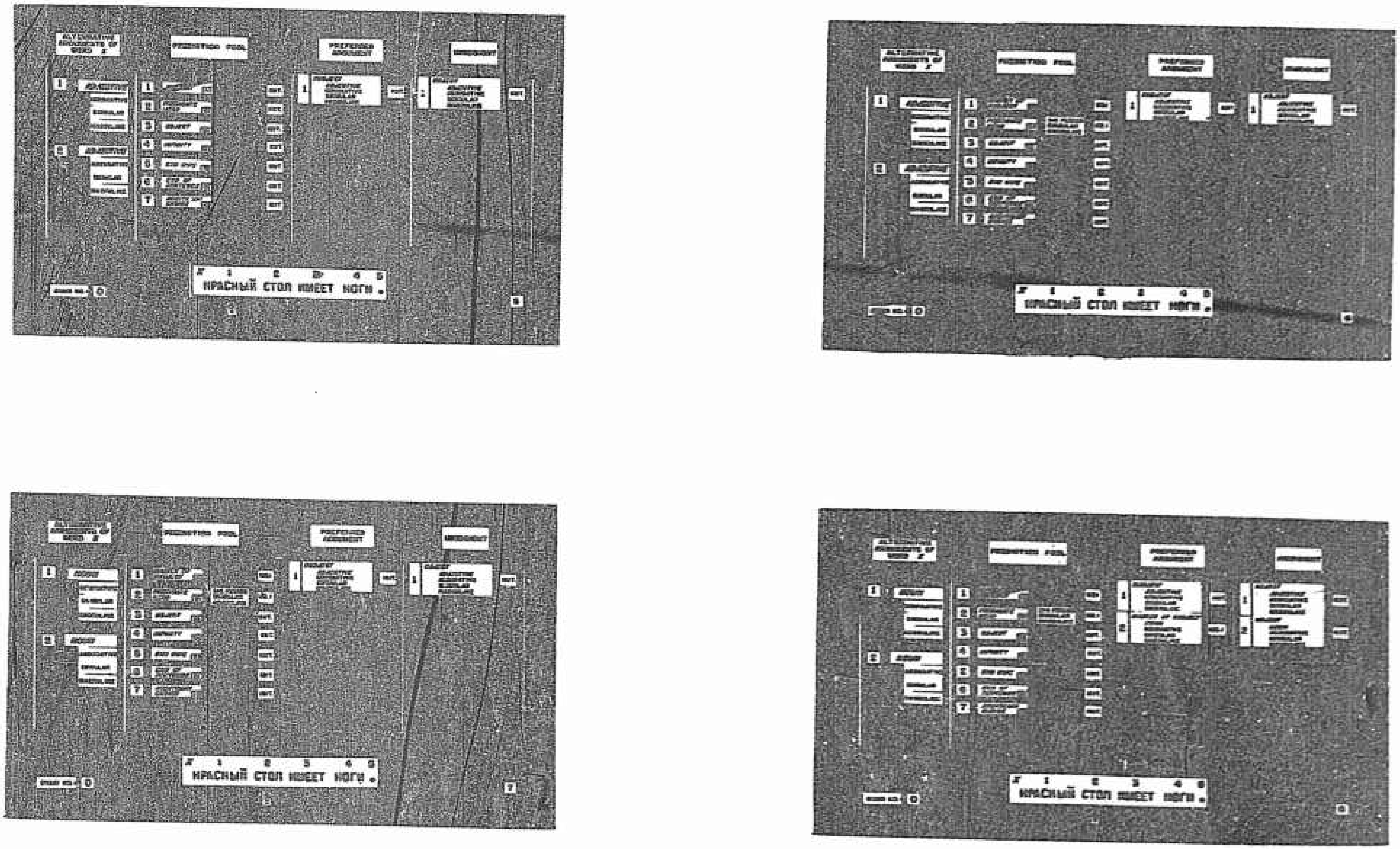


Fig. 5-1 (continued)

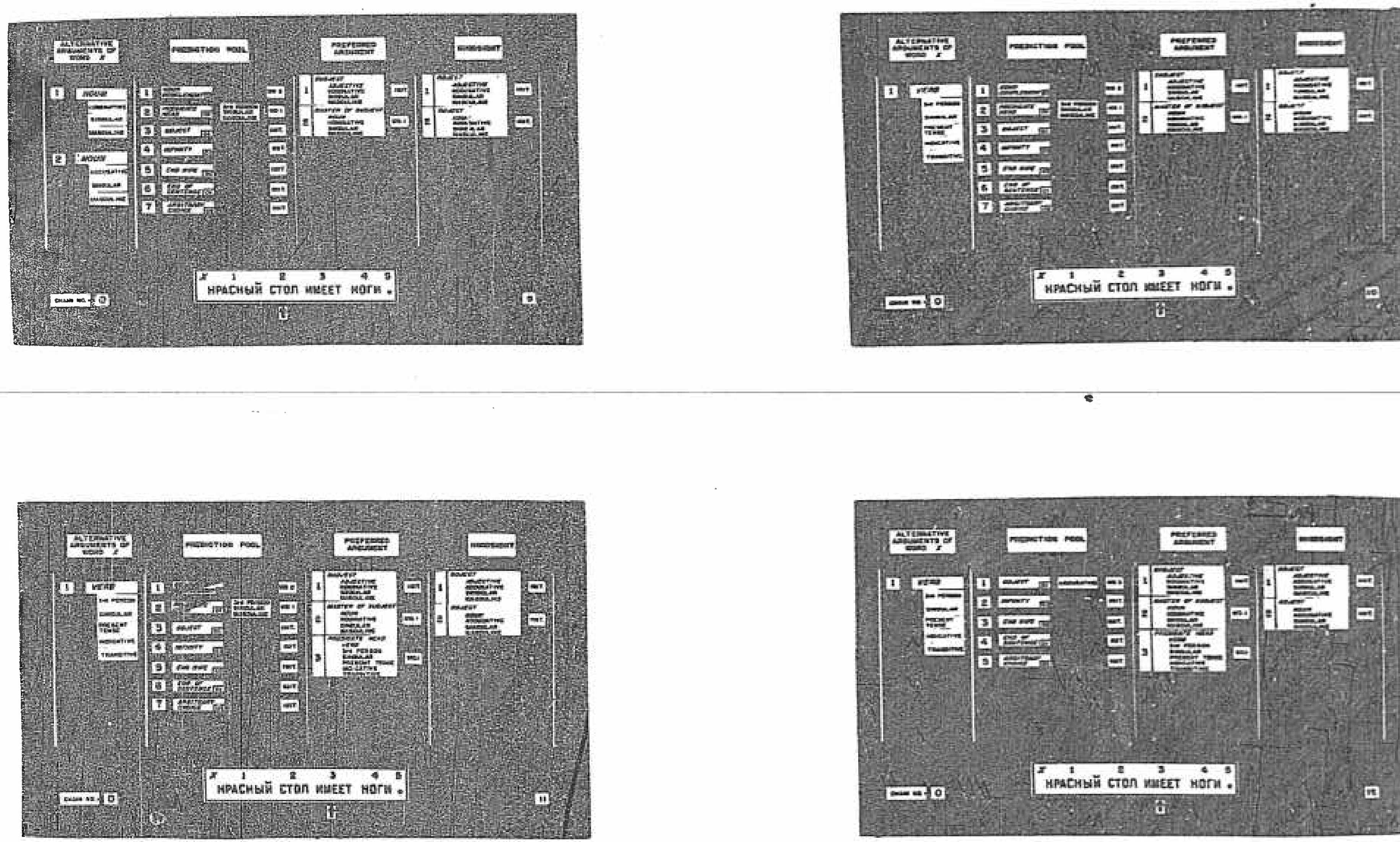
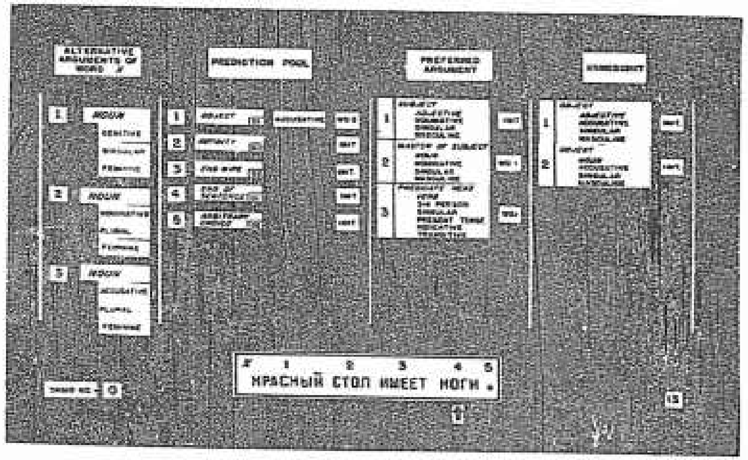
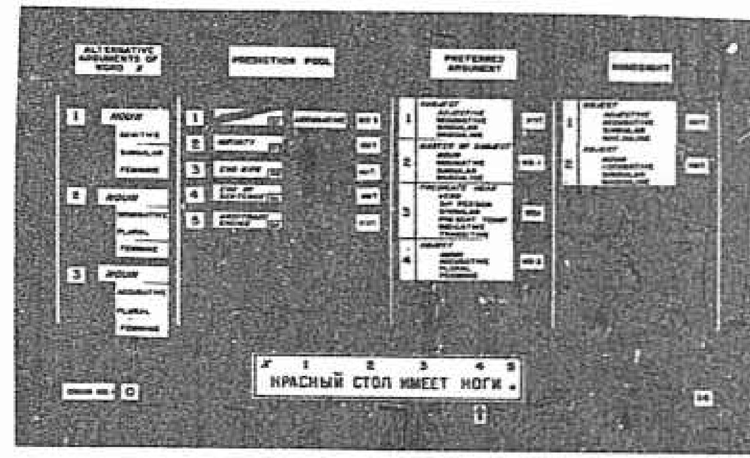
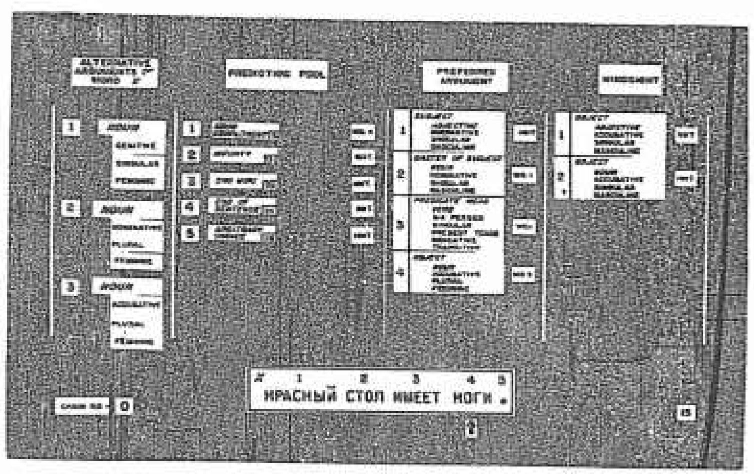


Fig. 5-1 (continued)







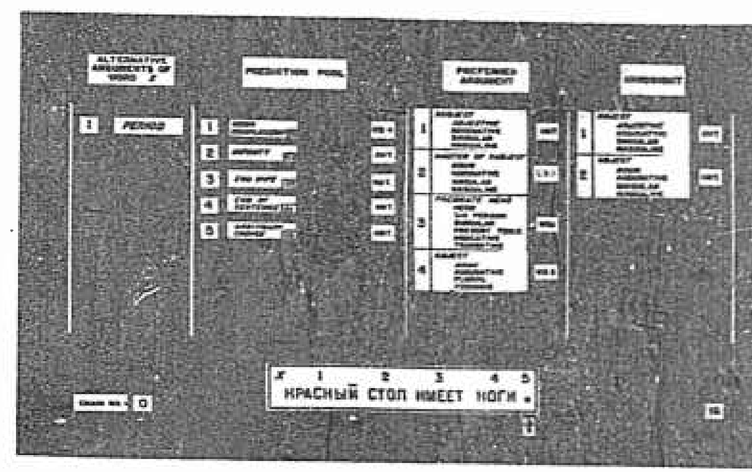
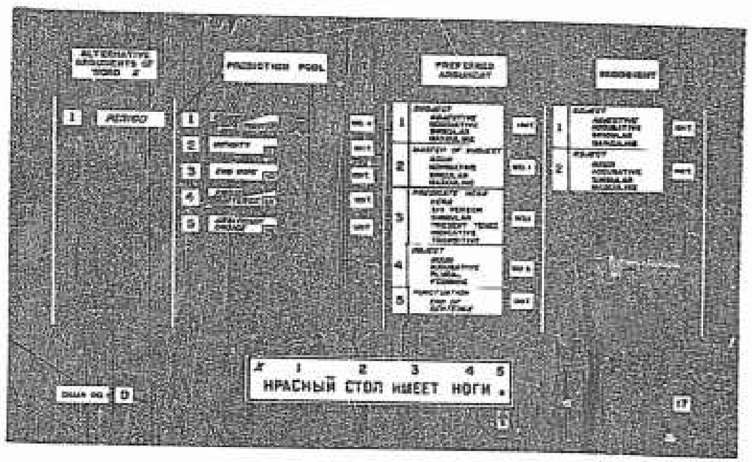


Fig. 5-1 (continued)



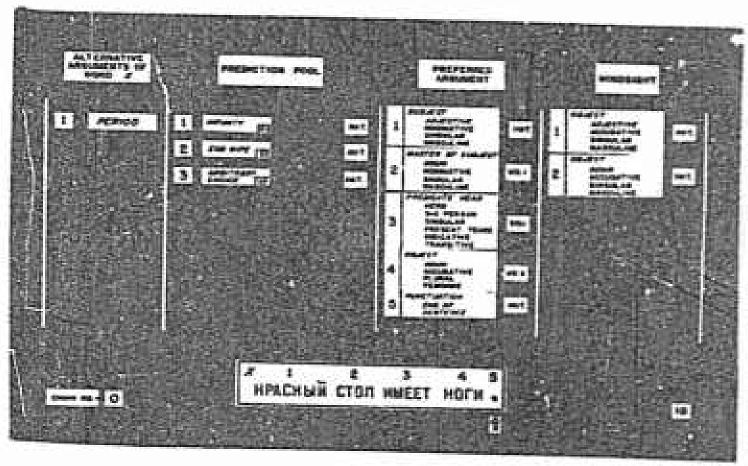


Fig. 5-1 (continued)

(3) Prediction span indicator (PSI) - A prediction span indicator is assigned to each prediction indicating how long the prediction is to be allowed to remain in the pool. The prediction span indicators used in the simplified illustration are:

PSI = 00 - The prediction must be fulfilled by the next word in sequence or not at all.

PSI = 01 - The prediction must be fulfilled during the analysis of the sentence.

PSI = 02 - The prediction may be fulfilled more than once in a single sentence and therefore must never be wiped (that is, erased) from the prediction pool.

These definitions of the prediction span indicators are intended solely for the illustration of the simplified program. New PSI definitions will be made when the present experimental program is discussed in detail (Section 4).

(4) Intersection - In testing the alternative arguments of a word against the predictions in the prediction pool, an intersection takes place when an alternative argument can fulfill a prediction.

(5) Preferred argument - The preferred argument is the alternative argument of the first intersection in a test sequence (see Definition 21, Section 4.4A). In the test sequence, all the alternative arguments of a word are tested against all the predictions in the pool in their respective orders, such that each prediction, in turn, is tested against the set of alternative arguments. The prediction that intersects with the preferred argument becomes known as the attributed argument of the word.

(6) Hindsight - During analysis, information, other than the preferred argument, that has to be stored, is put onto a second output file, called the hindsight. For example, if more than one alternative argument intersects with a prediction in the pool, all intersecting alternative arguments but the first, which is the preferred argument, are put into hindsight.

(7) Chain number - The chain number is an index that is incremented whenever the predictive syntactic analysis program cannot, on the basis of the predictions stored in the prediction pool, select a preferred argument for a word.

The first step in the program, at the beginning of each sentence, is to set the chain number to zero and insert an initial set of predictions into the prediction pool (Fig. 5-1.2). The PSI and the source of the prediction are stored with each prediction. The symbol "INIT." refers to the seven initial predictions. The four predictions with PSI = 01, subject, predicate head, object, and end of sentence, predict the corresponding elements of the sentence which, for the purpose of this example, are self-explanatory. The functions of the other three predictions will be discussed subsequently in Sections 3 and 5.

Each word is processed by the program in a three-step cycle: (1) the alternative arguments of the word are placed in a central memory location; (2) each prediction is tested against all the alternative arguments of the word, the preferred argument is identified and noted, and the appropriate information is recorded on hindsight; (3) the prediction pool is updated. An accurate syntactic analysis is closely tied to the

ordering of the predictions in the pool and of the alternative arguments of the words. The ordering of the alternative arguments is only secondary, however, since all the alternative arguments are tested against every prediction in turn.

After the alternative arguments of красный are brought into memory (Fig. 5-1.3), the testing for intersections begins. The first intersection is found in the test of the first alternative argument, /adjective, nominative, singular, masculine/, against the first prediction, subject. The preferred argument (Fig. 5-1.4) and the attributed argument, together with the source of the fulfilled subject prediction, are entered on the main output file (which is labeled in Fig. 5-1 "preferred argument"). The subject prediction is crossed out to indicate that, since it has been fulfilled, it will be wiped from the pool when the pool is updated.

The testing for intersections is continued. No intersections are encountered in the tests between the first prediction and the second alternative argument, the second prediction and either alternative argument, and the third prediction and the first alternative argument. A second intersection is discovered between the object prediction and the alternative argument, /adjective, accusative, singular, masculine/. Since the preferred argument has already been established, this intersection is recorded on the hindsight file (Fig. 5-1.5).

It is necessary to record the alternate intersections, since the selection of красный as the subject is made arbitrarily, based only on the ordering of the predictions in the pool. In the analysis of any sentence, there is no way of knowing whether the arbitrary selection is the correct one, without analyzing the remainder of the sentence. In the event it is

discovered later that the selection was made inappropriately, the hindsight will contain a list of the other possible alternatives which can be substituted for the inappropriate one.

The ordering of the predictions in the pool is of primary importance in the analysis of a sentence. The predictions that are expected to be fulfilled first in regular sentences are placed toward the top of the pool. Thus the subject prediction is above the predicate head prediction, which, in turn, is above the object prediction. If, at a given point in the analysis of a sentence, there is a choice of several predictions which might be fulfilled, then the most likely prediction will provide the first intersection.

After the second intersection, the testing for intersections is continued once more, but no more intersections are found. After the completion of the testing phase, the prediction pool is updated. The fulfilled subject prediction is wiped from the pool. Every adjectival preferred argument generates a master prediction with PSI = 00, where a master is defined as a noun or another adjective following immediately after the analyzed adjective and agreeing with the analyzed adjective in case, number, and gender (Fig. 5-1.6).

Also, after identifying the subject of the sentence, it is possible to modify the predicate head prediction, since the predicate must agree with the subject in person, number, and gender. In this particular example, the predicate head is modified so that only a third person, singular, masculine predicate can fulfill the prediction.

The source of both the master prediction and the modified predicate head prediction is listed as "WD 1", referring to the first word of the

sentence. The source of a modified prediction is always listed as the number of the last analyzed word that has modified the prediction.

The testing cycle for красный has been completed and a new cycle is started by bringing into memory the alternative arguments of the second word, the noun стол (Fig. 5-1.7).

Two intersections are found when testing the alternative arguments of стол against the predictions in the pool (Fig. 5-1.8). The preferred argument and attributed argument, due to the first intersection between the master prediction and the alternative argument, /noun, nominative, singular, masculine/, are recorded on the main output file. The second intersection between the object prediction and /noun, accusative, singular, masculine/ is posted on the hindsight file.

The prediction pool is then updated (Fig. 5-1.9). Every nominal preferred argument produces a noun complement prediction which can be fulfilled by an adjective or noun in the genitive case following immediately after the analyzed noun. The noun complement replaces the fulfilled master prediction at the top of the pool. Since there are no other modifications to the prediction pool, the alternative argument of the following word, the verb имеет, is brought into the central memory location (Fig. 5-1.10).

Only one intersection is discovered, resulting in the attributed argument, predicate head, and the preferred argument, /verb, third person, singular, present tense, indicative, transitive/ (Fig. 5-1.11).

In updating the prediction pool, the noun complement together with the predicate head is wiped, since the PSI of the former prediction is 00 and the prediction has not been fulfilled. Since the verb is transitive, the object prediction can be modified so that only an accusative object can

fulfill the prediction (Fig. 5-1.12). Prior to this modification, either an accusative object or an instrumental object would have been accepted.

After the prediction pool is updated, the three alternative arguments of the noun норм are brought into the central memory location (Fig. 5-1.13). The testing for intersections is then resumed.

There is a single intersection resulting in the attributed argument, <u>object</u>, and in the preferred argument, /noun, accusative, plural, feminine/ (Fig. 5-1.14). After this information is recorded on the main output file, the prediction pool is updated once again. Since the last analyzed word had a nominal preferred argument, a <u>noun complement</u> prediction is entered at the top of the pool. (Fig. 5-1.15).

The single alternative argument of the punctuation mark, /period/, is then brought into the central memory location (Fig. 5-1.16). Testing of the alternative argument against the predictions in the pool produces one intersection, which results in the preferred argument, /end of sentence/ (Fig. 5-1.17). The prediction pool is updated for the last time, and both the <u>noun complement</u> and the <u>end of sentence</u> predictions are wiped, the former because its PSI equals 00. The analysis is now complete (Fig. 5-1.18).

The results of this analysis will now be reviewed. For every word in the sentence a preferred argument has been selected according to the contents of the prediction pool. This is indicated by the fact that the chain number is still zero. No predictions with PSI = 01 remain in the prediction pool, which indicates that every prediction that was expected to be fulfilled was indeed fulfilled during the analysis of the sentence.

These two results, chain number equal to zero and no remaining predictions with PSI = 01, occurring together, give a strong indication that a correct syntactic analysis of the sentence has been obtained. This is not meant to imply that the analysis is both unique and correct. A stronger indication would exist if, in addition, there were no information recorded on the hindsight file. To determine whether another analysis is feasible, the entire analysis procedure must be repeated and the first word must be considered as the object of the sentence. In this example, of course, no alternative analysis is possible.

3. End Wipe and Arbitrary Choice Predictions

The analysis of the sentence, Красный стол имеет ноги, proceeded in a straightforward manner. The output of the program was a correct syntactic analysis, as a matter of fact, the only possible correct analysis. Such a simple sentence can always be correctly analyzed on a single pass.

The true merits of predictive syntactic analysis become evident only when the ability of the program to detect errors in analysis and to record clues for a projected correcting pass is considered. If it is assumed that (1) the sentence being analyzed is grammatically correct, so that there is no need to test whether or not the sentence is grammatical, but only to find a grammatical formulation of the set of alternative arguments, (2) all the words have been found in a dictionary in which there are no errors, and (3) the words in the sentence have not been misspelled, then the two predictions, end wipe and arbitrary choice, provide a mechanism for the detection of errors in the analysis. The rules for the operation of these two predictions in the existing program are as follows:

(1) End Wipe - If no intersection has been discovered in the testing of all the predictions located in the pool above the end wipe prediction against the set of alternative arguments of the word currently being tested, then all of the tested predictions, including the end wipe, are to be wiped from the prediction pool.

For the purposes of this simplified example, however, only such predictions that do not have a PSI = 02 will be wiped from the prediction pool. Since the end wipe prediction itself has a PSI = 02, it will not be wiped. So long as only a simple sentence is considered, the scheme adopted for this example cannot be distinguished from the one that is used in the experimental program.

(2) Arbitrary Choice - If no intersection has been discovered in the testing of all of the predictions located in the pool above the arbitrary choice prediction against the set of alternative arguments of the word currently being tested, then the first alternative argument of the word is to be selected as the preferred argument, the attributed argument arbitrary choice is to be assigned to the word, all other alternative arguments of the word are to be listed on the hindsight file, and the chain number is to be incremented.

The end wipe prediction serves a double purpose when used in the manner outlined. Primarily, it functions in the prediction pool as a sentinel designating the end of a set of predictions of a given nested structure in the sentence (see Section 4.3). Having reached this sentinel with no previous intersections, it is assumed by the program that the nested structure has been completely analyzed, and the word being analyzed

belongs to another nest in the sentence. This function of the <u>end wipe</u> prediction is not self-evident in the simple example of this section, but will be pointed out later when actual output of the predictive syntactic analysis program is studied.

The second function of the <u>end wipe</u> prediction is to provide a mechanism to wipe the entire prediction pool in the event an error is discovered. An error in analysis is assumed whenever there are no intersections between the alternative arguments of a word and the predictions in the pool. Since an error is always discovered after the fact, there is a question as to which predictions in the prediction pool might be meaningless because of the propagation of this error. Rather than leaving the predictions in the pool and continuing the possibility of propagating an error after its existence has been ascertained, the predictions in the pool with several exceptions are wiped and the analysis continues with a clean slate. The second function is actually a special case of the first function when all the predictions in the pool are considered as the nested structure of the sentence as a whole.

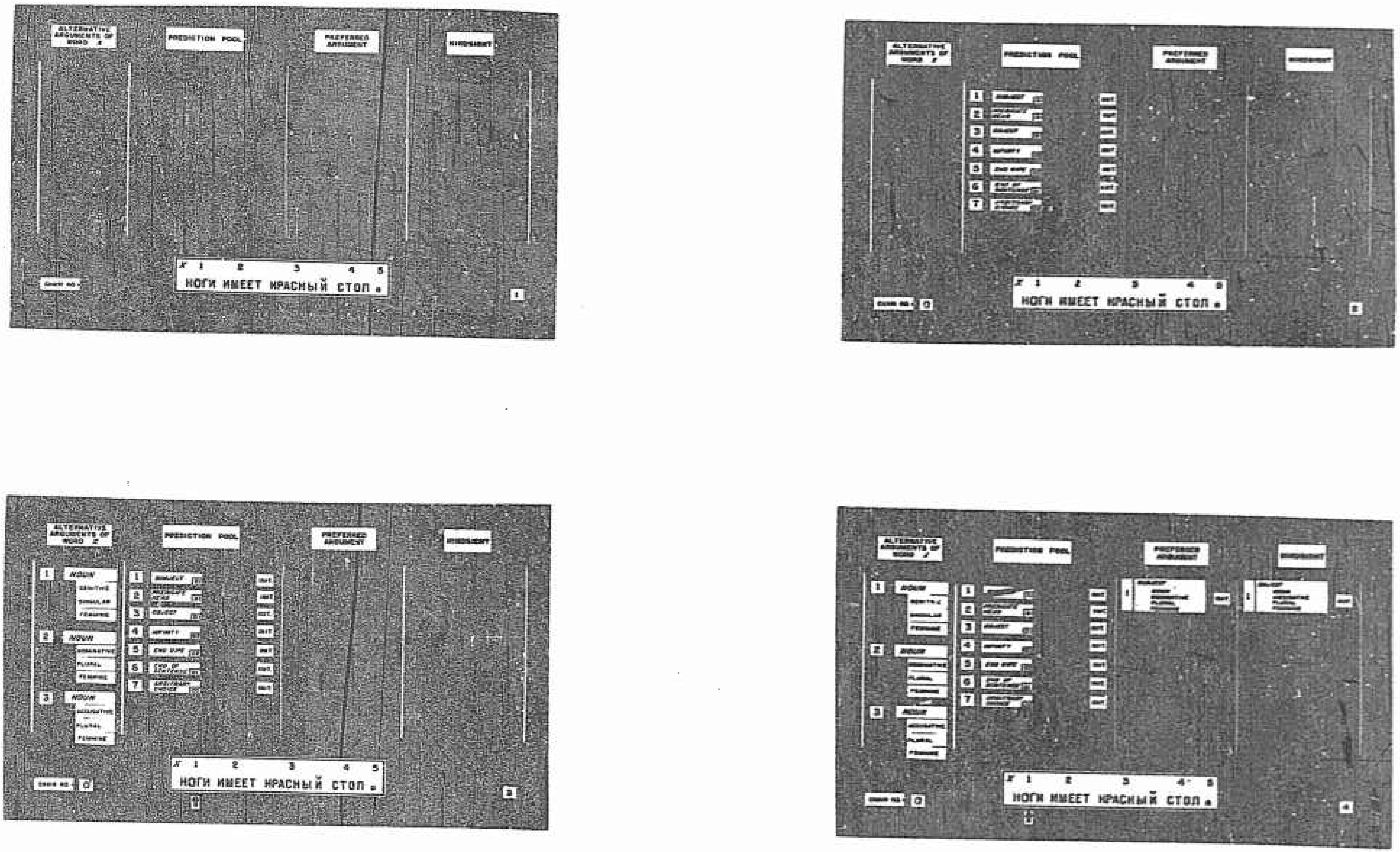
The significance of this wiping operation is that whereas any nested structure, the beginning of which has already been recognized and for which predictions have been made, will not be analyzed completely, complete nested structures, occurring to the right of the word which causes the wiping of the prediction pool, will be analyzed correctly. For languages in which a Δ_M-theorem holds this is true, as has been proven for certain artificial languages.

The predictive syntactic analysis method requires that a preferred argument be selected for every word. Even if the attributed argument is

an arbitrary choice, new predictions can be generated for the updated prediction pool by the preferred argument, and so any nested structure which can be predicted by the word labeled arbitrary choice can be identified on the same pass.

Since the first alternative argument is arbitrarily chosen as the preferred argument, in the event it is discovered later that this choice was made in error, a list of the other potential choices will be available in the hindsight file. If the alternative arguments are ordered in decreasing probability of occurrence, then arbitrary choice preferred arguments will have the best opportunity of being selected correctly. As was mentioned earlier, the ordering of the alternative arguments is only secondary, however, since all the alternative arguments are tested against each prediction in turn. In the instances where more than one intersection is found, the greatest effect on the selection of the preferred argument will be the ordering of the predictions in the prediction pool, as discussed in the preceding section. Poor ordering, especially in the prediction pool, will be indicated by frequent wrong analyses on the first pass of a sentence, with the correct analysis noted on the hindsight file.

The end wipe prediction in its second role and the arbitrary choice prediction are used in the second illustrative example (Fig. 5-2). The same prediction span indicators are used in this example as in the example of the preceding section. The same words in a rearranged order, corresponding to the emphatic statement: Ноги имеет красный стол, will be used (Fig. 5-2.1).



A Second Simplified Example of Predictive Syntactic Analysis

Fig. 5-2

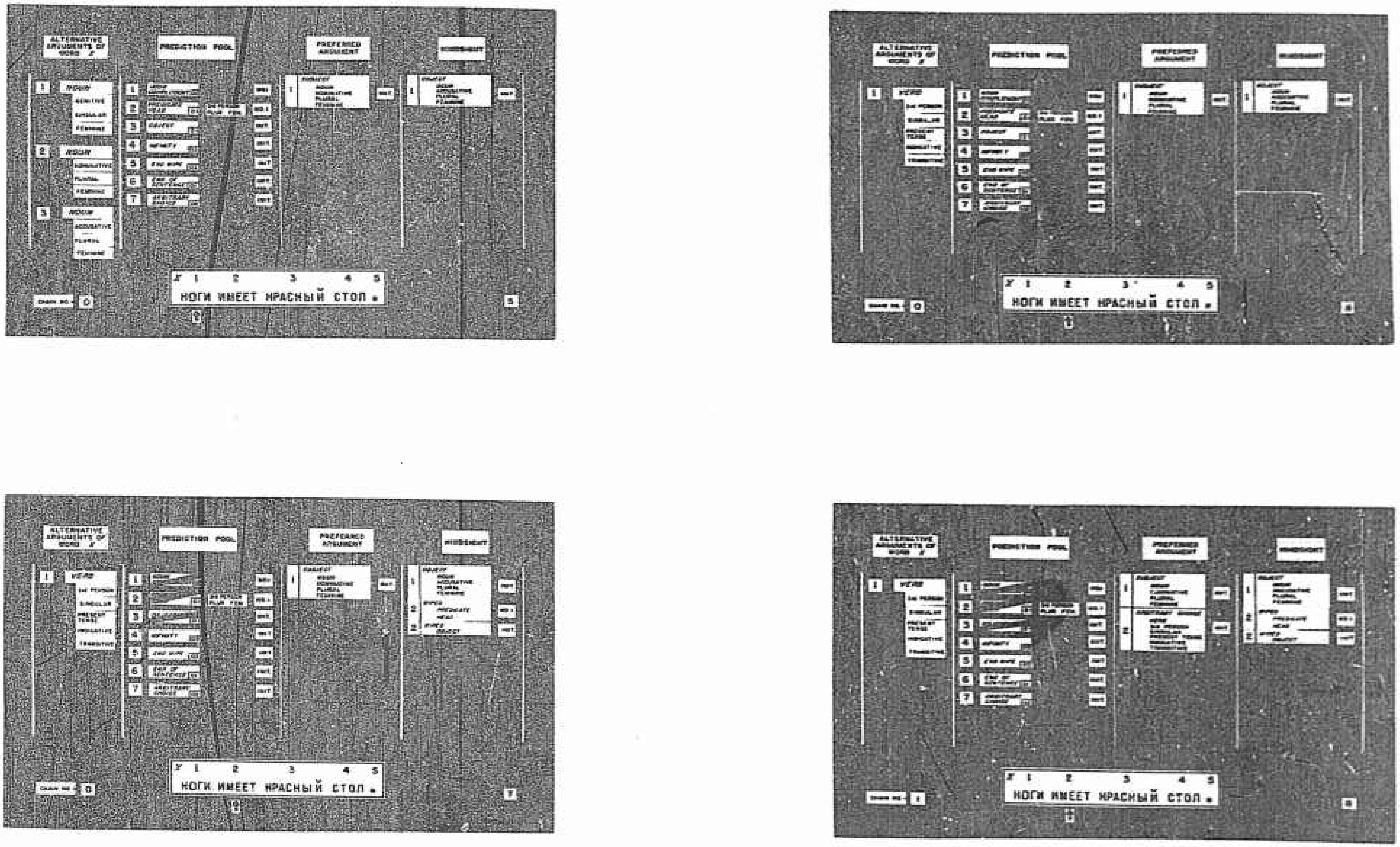


Fig. 5-2 (continued)

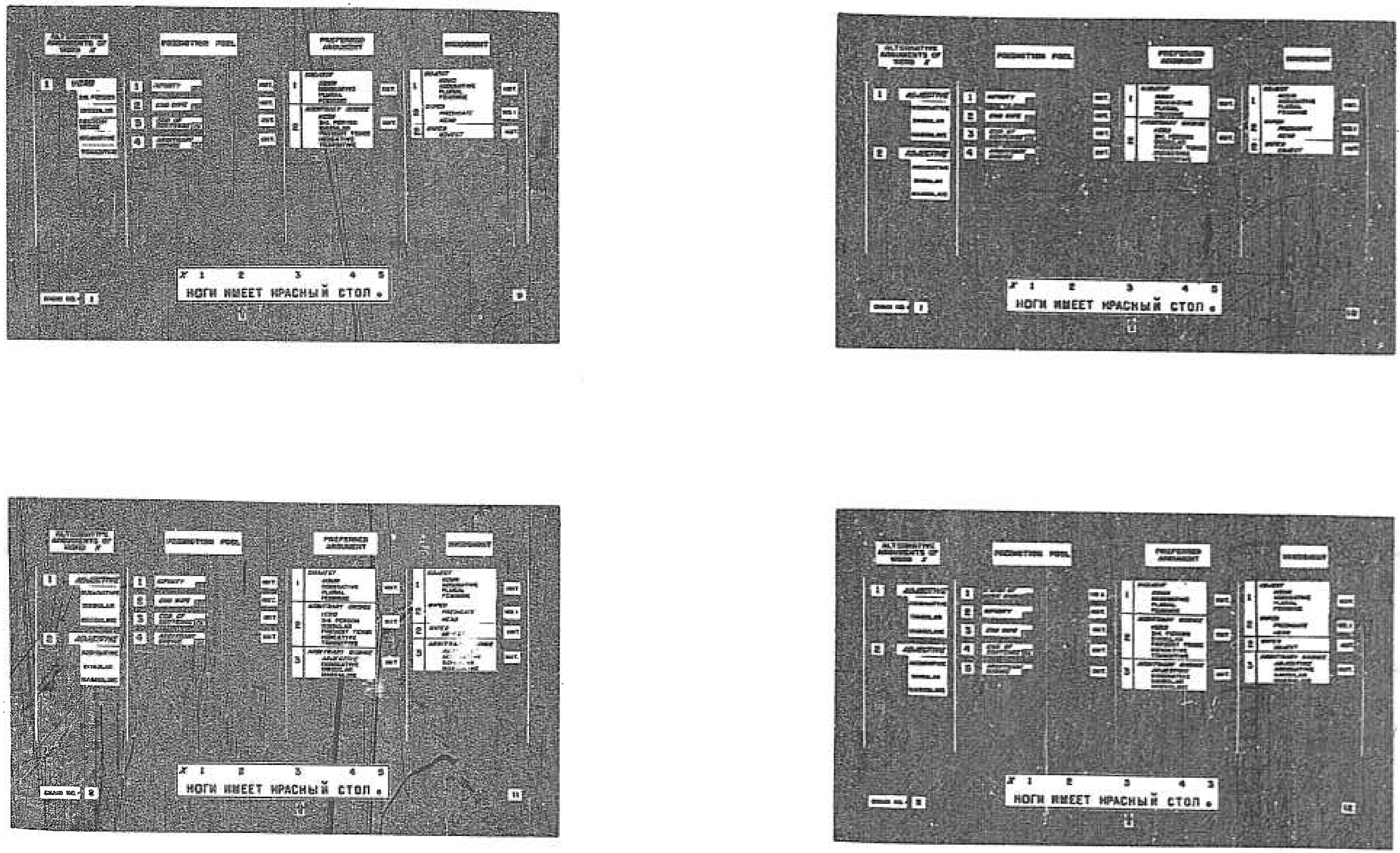


Fig. 5-2 (continued)

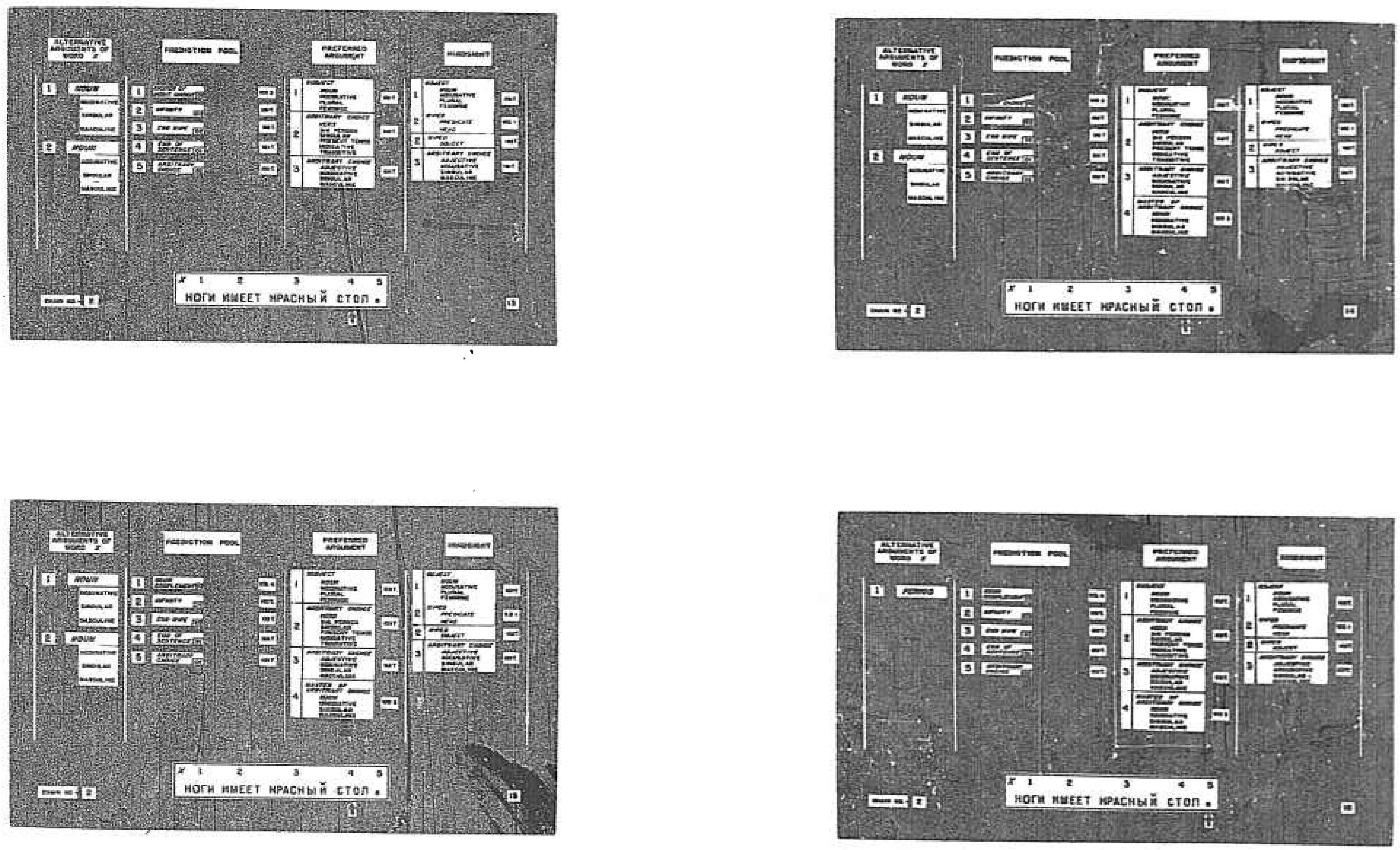
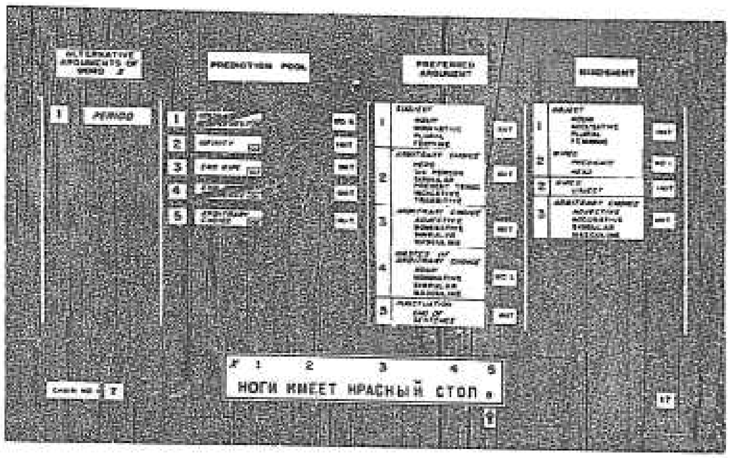


Fig. 5-2 (continued)



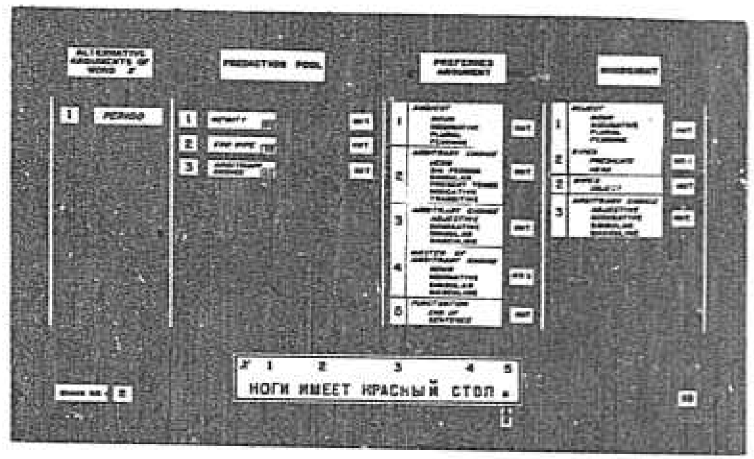


Fig. 5-2 (continued)

The analysis starts in the same manner as in the previous example. After initializing the program (Fig. 5-2.2), the alternative arguments of НОГИ are brought into the central memory location (Fig. 5-2.3). The attributed argument, subject, and the preferred argument, /noun, nominative, plural, feminine/, are assigned to НОГИ as a result of the first intersection between the first prediction and the second alternative argument (Fig. 5-2.4). The second and only other intersection between the object prediction and the alternative argument, /noun, accusative, plural, feminine/, is noted on the hindsight file.

The prediction pool is updated with the addition of the noun complement prediction after the subject prediction has been wiped (Fig. 5-2.5). Since the subject prediction has been fulfilled, the predicate head prediction can be modified so that only a third person, plural, and feminine predicate can fulfill the prediction.

The one alternative argument of the verb ИМЕЕТ is brought into the central memory location (Fig. 5-2.6) and is tested against the predictions in the pool. There is no intersection with the noun complement prediction. Likewise, there is no intersection with the predicate head prediction since ИМЕЕТ is singular and the prediction has been modified so that only a plural predicate can fulfill it. No intersections are discovered in testing the alternative argument against the object and infinity predictions. (The latter prediction will be discussed in Section 5.)

The lack of an intersection is sensed by the end wipe prediction, which then wipes some of the predictions from the prediction pool (Fig. 5-2.7). The predictions for which PSI = 02 are not wiped (by the definition adopted for this example). Since two of the predictions,

predicate head and object, that are wiped have PSI = 01, their wiping is recorded on the hindsight file. The testing for intersections is continued. Since there is no intersection with the end of sentence prediction, the arbitrary choice prediction selects the alternative argument as the preferred argument and assigns the attributed argument, arbitrary choice, to имеет. The arbitrary choice prediction also increments the chain number (Fig. 5-2.8).

Even though the verb is transitive, no object prediction which can be modified is left in the prediction pool. The remaining four predictions are pushed to the top of the prediction pool (Fig. 5-2.9). The two alternative arguments of красный are then brought into the central memory location (Fig. 5-2.10).

Once more, no intersections have been found when the end wipe prediction is being tested. But since there are no predictions in the pool that can be wiped, there is no explicit change in the pool. No intersections have been found when the arbitrary choice prediction is being tested, so that the attributed function, arbitrary choice, is assigned to красный. Since there are two alternative arguments of красный, the first one is arbitrarily selected as the preferred argument, and the second one is recorded on the hindsight file (Fig. 5-2.11).

A master prediction is entered at the top of the updated prediction pool since красный has an adjectival preferred argument (Fig. 5-2.12). The alternative arguments of стол are brought into the central memory location (Fig. 5-2.13) and are tested against the predictions in the pool. A single intersection is discovered which results in the attributed argument, master (of arbitrary choice), and the preferred argument, /noun, nominative, singular, masculine/ (Fig. 5-2.14).

The prediction pool is updated with the addition of a noun complement prediction (Fig. 5-2.15), and the alternative argument of the punctuation mark is brought into the central memory location (Fig. 5-2.16). The single intersection resulting in the preferred argument, /end of sentence/, is noted on the main output file (Fig. 5-2.17), after which the prediction pool is updated for the last time (Fig. 5-2.18). The noun complement prediction is wiped at this time because its PSI = 00.

If the output is now scanned, the chain number is discovered not to be equal to zero, and if the sentence is, indeed, grammatically correct, it can be assumed that there was an error in the analysis. In the analysis of this sentence, the error can be identified in the hindsight by the alternate object attributed argument of ноги and the wiped object prediction. A second pass through the sentence, assigning the alternative attributed argument to ноги, would lead to a correct syntactic analysis.

Although in the analysis of the first word of the sentence there was an error which was subsequently discovered when analyzing the second word, no attempt was made to correct the error at that time. In this sentence the error was obvious and could have been corrected immediately. But it is possible that errors in other sentences might not be so obvious, and there might be several clues throughout the remainder of the sentence that would aid in determining the necessary correction. While continuing with the analysis, the subordinate nested structure of the noun phrase, красный стол, was correctly identified, as would be any other nested structure that followed in its entirety the identification of the error. Unless some evidence suggesting that corrections be made at once when the errors are

discovered comes to light, correction will be attempted only after the analysis of an entire sentence.

Since the implications involved in error correction are not yet clear or understood, no attempt has been made yet to write such a program.

To conclude the discussion of this illustration, it is interesting to see what would have happened if the end wipe and arbitrary choice predictions had not been invoked and the noun complement, predicate head and object predictions had been allowed to remain in the pool after the error was discovered. Имеет would have modified the object prediction so that only an accusative object would have fulfilled the prediction. The adjective красный would have been accepted as the object of имеет, and стол would have been accepted as the master of (the accusative adjective) красный. This result seems to be far less satisfactory than the one illustrated.

4. The Predictive Syntactic Analysis Program

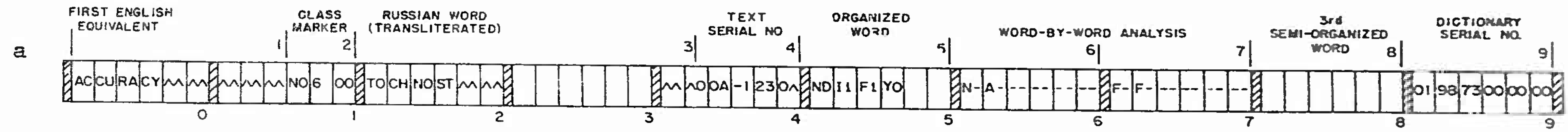
The input to the predictive syntactic analysis program is a text, in which every word is represented by a line in the texthadic format (Fig. 5-3a) (see Section 3.4). Two outputs, the main output file (Fig. 5-3b) and the hindsight file (Fig. 5-3c), are produced by the program. Column 9, which in the texthadic format contains the dictionary entry number, is replaced on both output files by the attributed argument of the word and by the text serial number (modulo 1000) of the word that was the source of the prediction that resulted in the attributed argument. In columns 6 and 7 of the output file, the alternative arguments are replaced by the preferred argument. On the hindsight file, each intersecting alternative argument

that has not been selected as the preferred argument is represented by a line, and the alternative argument itself is placed in columns 6 and 7. Two extra columns exist on the main output file which are referred to as columns 3A and 3B. Column 3A contains the chain number after the analysis of the word represented by the 10-word item. Column 3B contains the number of predictions in the prediction pool before the analysis of the current word. Moreover, whenever a prediction which should have been fulfilled is wiped from the pool, it is marked on the hindsight file (Fig. 5-3d).

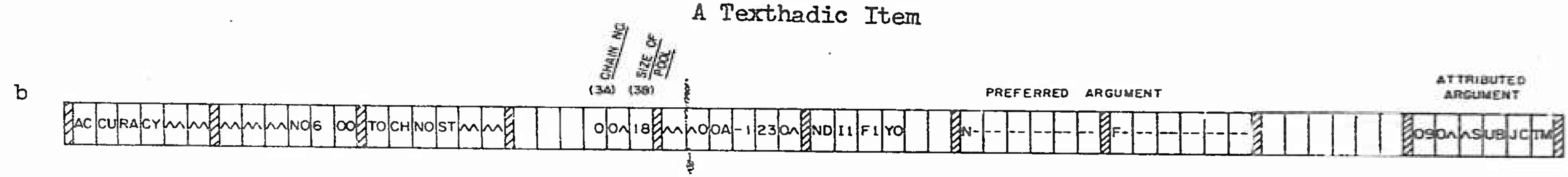
It should be stressed once more that the single English correspondent of the Russian word that is included in a texthadic item has little significance in the translation of the examples given in this chapter. The purpose of its appearance is to aid the reader who understands no Russian.

The machine program that has been written by Bossert consists of two sets of subroutines in addition to a skeletal section. The actual analysis is carried out by the subroutines while the skeletal section performs the necessary bookkeeping tasks. The skeleton provides the mechanism for stepping through both the predictions in the pool and the alternative arguments, so that a single alternative argument is tested against a single prediction at a time. It also provides the mechanism for updating the prediction pool.

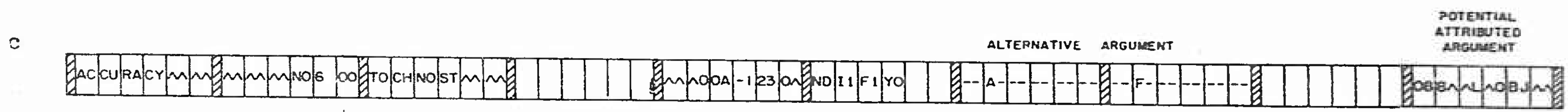
The first set of 22 subroutines, called essences (Table 5a of Appendix F) represent syntactic relationships that are predicted and fulfilled during syntactic analysis. The subroutines themselves carry



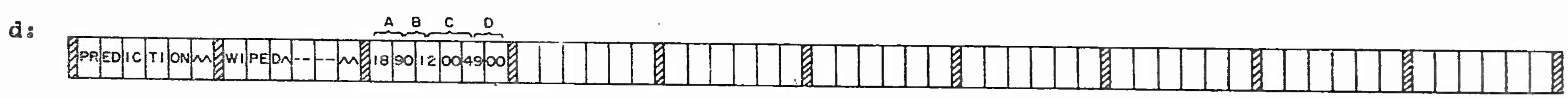
A Texthadic Item



A Preferred and Attributed Argument



A Hindsight Item



A = LAST THREE DIGITS OF SERIAL NUMBER OF WORD MAKING PREDICTION.
B = PSI
C = PROGRAMMING INFORMATION.
D = WIPED PREDICTION.

900	SUBJECT	927	OBJECT
910	LEFT OBJECT	945	PREPOSITION COMPLEMENT
915	PREDICATE HEAD	972	MASTER

An Indication of a Wiped Prediction

Output Format of the Experimental Predictive Syntactic Analysis Program

Fig. 5-3

out all the tests to determine whether a prediction is fulfilled by one (or more) of a set of alternative arguments. There is an essence subroutine for every prediction that can be stored in the prediction pool.

The second set of 25 function type subroutines (Table 5b of Appendix F) represent word categories similar to the familiar "parts of speech" and "syntactic roles". These subroutines make the new predictions which are then put into the prediction pool and also modify existing predictions in the pool. The first group consists of 15 subroutines that represent the parts of speech and make new predictions based on the preferred arguments of the analyzed words, whereas the second group consists of 10 subroutines that represent the syntactic roles and modify existing predictions in the pool according to the attributed arguments of the analyzed words.

If the name of a subroutine is likely to be misleading, a suffix "-E" for essence type subroutine and a suffix "-T" for the function type subroutine have been appended to the name.

The subroutines are completely independent, that is, only one subroutine is used at a time. Once control is passed to the subroutine, the subroutine retains control until the testing or generating process is completed, after which control is returned to the skeletal program. The relationships among the alternative arguments, the predictions, and the subroutines are shown in the tables of Appendix F. A detailed example of the use of the tables is also given in the appendix.

The interrelationships between the predictions and the alternative arguments have been condensed and summarized so that they could be

presented, in their entirety, on two pages (Table 5-1 and 5-2). The preferred arguments that can fulfill the 22 essences (or predictions) listed in Table 5-1. In Table 5-2 are listed the predictions that are made or modified by the preferred and attributed arguments.

As an example of the use of these tables, the subject prediction can be fulfilled by a noun, pronoun, adjective, numeral, or verb alternative argument (Table 5-1). This table does not indicate that the first four alternative arguments must be nominative, nor does it indicate that the verb must be infinitive. For this detailed information, the tables in Appendix F must be referred to. If a noun is selected as the subject, the noun complement prediction is made by the noun preferred argument, the predicate head prediction is modified, and a compound subject prediction as well as an infinity and end wipe prediction is made by the adjective-noun subject attributed argument (Table 5-2).

With the set of subroutines that are in the experimental program, the following nested structures are recognized:

1. Noun structure - a string of adjectives terminated by a single noun, where all the adjectives and the noun agree in case, number, and gender.
2. Noun phrase - a noun structure in any case possibly followed by one or more noun structures in the genitive case.
3. Prepositional phrase - a preposition followed by a noun phrase where the initial noun structure is in a case that can be governed by the preposition.

Essences	Noun	Pronoun	Adjective	Numeral	Verb	Adjective Predicate Head	Participle	Preposition	Adverb	Infinite Conjunction	Comma	$------$	End of Sentence	Relative Conjunction-T	Gerund
Left Object-E	1	1	1	1											
Compound Left Object-E	1	1	1	1											
Object-E	1	1	1	1											
Compound Object-E	1	1	1	1											
Master/ (of essence)	1	1	1	1											
Noun Complement-E	1	1	1	1											
Compound Noun Complement-E	1	1	1	1											
Preposition Complement-E	1	1	1	1											
Compound Preposition Complement-E	1	1	1	1											
Arbitrary Choice	1	1	1	1	1										
Subject-E	1	1	1	1	1									1	1
Compound Subject-E	1	1	1	1	1										
Verb Master-E					1										
Compound Verb Master-E					1										
Predicate Head					1	1									
Compound Predicate Head					1	1									
Phraser					1		1								
Infinity								1	1	1	1	1			
End of Sentence-E													1		
Relative Conjunction-E														1	
Relative Pronoun-E															
End Wipe															

Alternative Arguments that Fulfill the Predictions in the Pool

TABLE 5-1

		Essences																					
		Noun Complement-E	Master/ (essence)	Preposition Complement-E	Object-E	Verb Master-E	Left Object-E	Subject-E	Predicate Head	Compound Predicate Head	Compound Subject-E	Compound Left Object-E	Compound Object-E	Compound Noun Complement-E	Compound Preposition Complement-E	Compound Verb Master-E	Phraser	Relative Conjunction-E	Relative Pronoun-E	Infinity	End Wipe	End of Sentence-E	Arbitrary Choice
Preferred Arguments	Adverb																						
	Numeral																						
	Pronoun																						
	Noun	1																					
	Adjective		1																				
	Preposition			1																			
	Verb				1	1																	
	Participle				1	1														1	1		
	Gerund																			1	1		
Attributed Arguments	Verb Predicate Head						m	m		①													
	Adjective Predicate Head						m	m		①													
	Adjective-Noun Subject								m		①									1	1		
	Pronoun Subject								m		①									1	1		
	Verb Subject								m		①									1	1		
	Left Object-T								m			①								1	1		
	Object-T												①							1	1		
	Noun Complement-T													①						1	1		
	Preposition Complement-T														①					1	1		
	Verb Master-T															①							
	Infinite Conjunction									a	a	a	a	a	a	a							
	Relative Conjunction-T																			a			
	Comma						1	1	1								1	1	1	1	1		
	Initial						1	1	1								1	1	1	1	1	1	1
	$------$																						
	End of Sentence																						

Key: 1 - Prediction made

① - "Compound" prediction made

m - Prediction modified

a - Prediction activated

Predictions Made by Preferred Arguments and Attributed Arguments

TABLE 5-2

4. <u>Verb phrase</u> (including participial phrase) - a verb or participle in any mood followed by one or more noun phrases in cases which can be governed by the verb or participle.

5. <u>Clause</u> - independent and dependent clauses are both treated in the same manner in the present program. Only three fundamental elements of a clause are considered: subject, predicate, and object. Usually there are several phrase structures within a clause.

The nested structures in the sentence often include combinations of clauses and the several types of phrases. All the efforts until now have concentrated on identifying all the members of a given clause or phrase so that, at this time, there is no scheme in the program to determine the syntactic relationships among the phrases and clauses.

The steps of the experimental predictive syntactic analysis program parallel quite closely the steps in the algorithms of the preceding chapter. The individual steps of the program are summarized formally in Iverson's notation (Program 5-1).

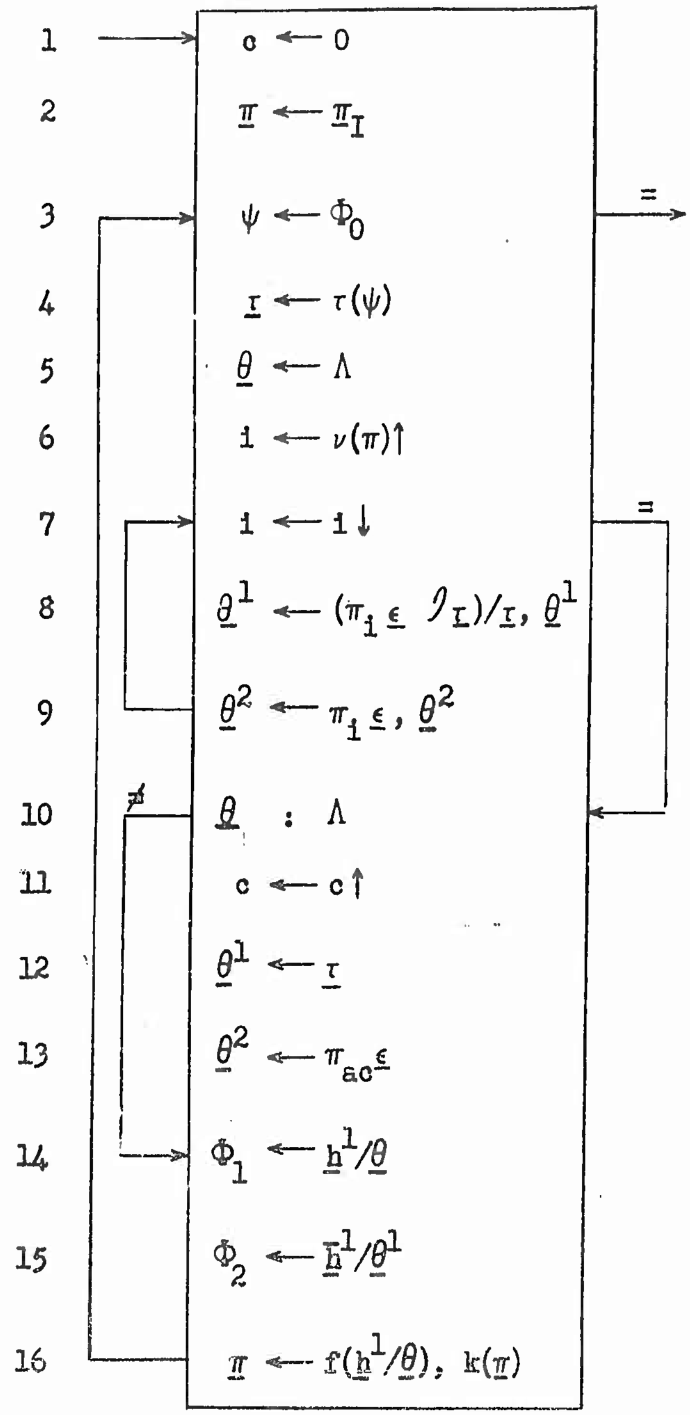
The program is initialized for each sentence in steps 1 and 2. The chain number is set to zero and an initial set of predictions is stored in the prediction pool.

The first word on the input file is read into the temporary store ψ, and the alternative arguments of the word are listed in $\underline{\tau}$ (Steps 3 and 4). A matrix $\underline{\theta}$, in which will be recorded all the possible pairs of preferred argument and attributed argument, is cleared in step 5.

Φ_0	Input file - sentence to be analyzed
Φ_1	Output file - preferred arguments and attributed arguments
Φ_2	Output file - hindsight
c	Chain number
$\underline{\pi}$	Vector representing predictions in prediction pool
$\underline{\pi}_I$	Set of predictions put into the pool at the beginning of every sentence
π_{ac}	Arbitrary choice prediction
$\underline{r}(x)$	Alternative arguments of word x
f(x)	Predictions to be made based on the preferred and attributed arguments of word x
$k(\underline{\pi})$	Updating operation on prediction pool

Symbols of Algorithm for Predictive Syntactic Analysis

TABLE 5-3



Algorithm for Predictive Syntactic Analysis

Program 5-1

The index i is initialized in step 6 and decremented in step 7 to allow for the process of step 8, where all the alternative arguments are tested against each prediction in the order of the listing of the predictions in the pool. This testing for intersections ("$\cap$") results in a logical vector with length equal to $L(\underline{\tau})$, each component of the vector equal to "1" if the corresponding alternative argument of $\underline{\tau}$ can satisfy the prediction π_i. The vector $\underline{\tau}$ is then reduced by this logical vector, and the corresponding potential preferred arguments are stored in $\underline{\theta}^1$. For each potential preferred argument, the appropriate potential attributed argument is stored in $\underline{\theta}^2$ (Step 9).

When this process has been carried out for each prediction in the pool, the program checks whether any preferred arguments have been discovered (Step 10). If not, the chain number is incremented in step 11 to indicate that there has been an error in the analysis, all the alternative arguments are transferred to $\underline{\theta}^1$ (Step 12), and the arbitrary choice attributed argument is placed into the corresponding positions of $\underline{\theta}^2$ (Step 13), so that the program can arbitrarily choose a preferred argument.

In step 14, the first alternative argument of $\underline{\theta}$, which is the first alternative argument intersecting with a prediction in the pool, is taken as the preferred argument. If no prediction has been fulfilled, the first alternative argument on the list $\underline{\theta}$ is recorded as the preferred argument on Φ_1 (Step 14). In either case, the appropriate attributed argument is also recorded on Φ_1.

All the other alternative arguments on the list $\underline{\theta}$ are stored in the hindsight file Φ_2 (Step 15). In the last step, new predictions are inserted

at the top of the prediction pool based on the preferred argument and the attributed argument of the analyzed word. The old prediction pool is appended to the new pool from below after suitable modifications, including the wiping of predictions due to the activation of end wipe, have been made to the predictions in the old pool.

The process returns to step 3 and the next word is read into ψ.

5. The Prepositional Phrase

The occurrence of certain words in a sentence such as adverbs, commas, and some prepositions cannot be predicted. They occur without any previous signal and therefore it is necessary to provide a special scheme to analyze such words. In the experimental program, the infinity prediction is the mechanism that permits the identification of such words independent of preceding words in a sentence. Since there is an infinity prediction in the pool at all times, these words are always predicted.

If a word is predicted by the infinity prediction, the syntactic structure of the sentence is incomplete. All that is known about the word is the nest within which it belongs, since each infinity prediction is located in a set of predictions in the pool representing a nested structure of the sentence under analysis. Only after the entire nest has been analyzed can the word predicted by infinity be tied syntactically to the rest of the nested structure. The infinity prediction always inhibits the action of the end wipe and arbitrary choice predictions, since it is located above the other two predictions in the pool.

Three examples will be used as illustrations of the analysis of prepositional phrases. The texthadic input, the main output file containing the preferred and attributed arguments, and the hindsight file (if any) will be shown with each example.

A straightforward analysis is illustrated by the phrase на анодной нагрузке лампы (Fig. 5-4). The rules for the analysis are given in Appendix F. The single alternative argument, /preposition/, of на fulfills an <u>infinity</u> prediction, at least one of which is always in the prediction pool. Since no other intersection is possible, nothing is written on the hindsight file.

A <u>preposition complement</u> prediction is made for every case and number combination that the preposition can govern. The four combinations that на can generate are indicated in column 6 of the texthadic item. The priority list for the ordering of the <u>preposition complement</u> predictions is given by the first three characters of column 8. In this instance, the prepositional (locative) predictions are listed prior to the accusative predictions. The singular prediction is always predicted prior to the plural prediction. The first few predictions in the pool after the analysis of на are:

1. Preposition complement (locative singular)
2. Preposition complement (locative plural)
3. Preposition complement (accusative singular)
4. Preposition complement (accusative plural)
5. etc. (old predictions)

Two predictions for each case are made for historic reasons only. It was convenient originally to make separate predictions for each case

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT	3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
ON	I01.00	N-A	00	25	00A-1504	P	--A--P--A--P	PAOROODF0560	INF PREP
ANODIC	A02.00	ANODN-OJ	00	28	00A-1505	AD00000	-----P------ -----F------		594 R COMP
LOAD	N04.31	NAGRUZK-E	00	28	00A-1506	ND11F000	-----P------ -----F------		595 R COMPM
TUBE	N04.00	LAMP-Y	00	28	00A-1507	ND12F000	-G---------- -F----------		596 N COMP

TEXTHADIC

				CODING DUE TO WORD-BY-WORD ANALYSIS		DICTIONARY SERIAL NO.
ON	I01.00	N-A	00A-1504	R	--A--P--A--P	PAOROODF0560	110780000000
ANODIC	A02.00	ANODN-OJ	00A-1505	AD00000	-G-CIP------ -F-FFF------		003090000000
LOAD	N04.31	NAGRUZK-E	00A-1506	ND11F000	---C-P------ ---F-F------		111900000000
TUBE	N04.00	LAMP-Y	00A-1507	ND12F000	-G----N-A--- -F----F-F---		098680000000

A Prepositional Phrase

Fig. 5-4

and number combination. To reduce the number of predictions in future versions of the experimental program, only two predictions should be made, one for each case. Then each prediction would accept either a singular or a plural prepositional complement.

The four alternative arguments of аходной are brought into a central memory location and are tested against the predictions for intersections. There is only one intersection between the first four predictions and the alternative arguments, resulting in the preferred argument /adjective, locative, singular, feminine/ and the attributed argument preposition complement. There are no other intersections with the previous predictions in the pool (from the earlier words in this sentence), so nothing is recorded on the hindsight file.

Since the PSI of the preposition complement prediction is 00, the three predictions which have not been fulfilled are wiped from the pool. Four new predictions are inserted at the top of the new pool, the master prediction by the adjectival preferred argument and the other three by the preposition complement attributed argument, in the following order:

(1) Master (of preposition complement) (locative, singular, feminine)

(2) Compound preposition complement (locative)

(3) Infinity

(4) End Wipe

(5) etc. (old predictions)

The two alternative arguments of нагрузке are brought into the central memory location. Once more there is only a single intersection between the alternative arguments and the predictions in the pool, and нагрузке is assigned the preferred argument /noun, locative, singular, feminine/ and the attributed argument master. Nothing is recorded on the hindsight file.

Since the master attributed argument makes no predictions, only the prediction of noun complement is made by the preferred argument. This prediction replaces the fulfilled master prediction at the top of the pool as follows:

(1) Noun complement

(2) Compound preposition complement (locative)

(3) Infinity

(4) End Wipe

(5) etc. (old predictions)

When the three alternative arguments of лампы are tested against the predictions, the only intersection results in the preferred argument /noun, genitive, singular, feminine/ and the attributed argument noun complement. Once more nothing is written on the hindsight file.

Several interesting points of this analysis are worth noting:

(1) All the predictions for the analysis of the prepositional phrase were located above the predictions that were in the pool just before the phrase occurred. In fact, there is an end wipe prediction located between the old predictions in the pool and the remaining new predictions.

The analysis of the phrase has been carried out entirely independently of any previous analysis of the sentence.

(2) All ambiguities in the adjective and the two nouns have been completely resolved, and a unique case and number has been assigned to each word.

(3) The analysis of the prepositional phrase was completed with no arbitrary choices, and no alternatives were recorded on the hindsight file. This would indicate that the analysis was carried out correctly and no other analysis could have been possible.

(4) The prepositional phrase consists of the preposition на and the two noun structures анодной нагрузке and лампы which together make up a noun phrase.

In contrast to the simple analysis of this phrase, consider the phrase в последующих каскадах (Fig. 5-5). The preposition в, which fulfills the infinity prediction, makes four preposition complement predictions as did на in the previous example, except that they are listed in the opposite order, accusative first and locative second, since the priority order in column 8 is different.

There are several intersections between the alternative arguments of последующих and the predictions in the pool. The first intersection is between the alternative argument, /adjective, accusative, plural/, and the accusative plural preposition complement prediction. The second intersection is between the alternative argument, /adjective, locative, plural/, and the locative plural preposition complement prediction. As usual, the alternative

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD		PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
IN(TO)	I01.00	-V	05	45	00A-0137	R		--A--P--A--P		APOROOBAU650	INF PREP
FOLLOWING	A04.00	POSLEDUJUSHC H-IX	05	49	00A-0138	AD0100	4	--------A---	---------A---		137 R COMP
STAGE	N01.00	KASKAD-AX	06	49	00A-0139	NDI1M000		-----------P	------------M		III ARBTR

TEXTHADIC

								CODING DUE TO WORD-BY-WORD ANALYSIS			DICTIONARY SERIAL NO.
IN(TO)	I01.00	-V			00A-0137	R		--A--P--A--P		APOROOBAU650	000020000000
FOLLOWING	A04.00	POSLEDUJUSHC H-IX			00A-0138	AD0100	4	-------6A--P	--------AA--A		152760000000
STAGE	N01.00	KASKAD-AX			00A-0139	NDI1M000		-----------P	------------M		086670000000

HINDSIGHT

								ALTERNATIVE ARGUMENT			POTENTIAL ATTRIBUTED ARGUMENT
FOLLOWING	A04.00	POSLEDUJUSHC H-IX			00A-0138	AD0100	4	-----------P	------------A		137 R COMP
FOLLOWING	A04.00	POSLEDUJUSHC H-IX			00A-0138	AD0100	4	-------6A--P	--------AA--A		136 FRASER
FOLLOWING	A04.00	POSLEDUJUSHC H-IX			00A-0138	AD0100	4	-------6A--P	--------AA--A		132 FRASER
PREDICTION WIPED	----	138013000972									

A Prepositional Phrase

Fig. 5-5

argument of the first intersection is assigned as the preferred argument, and the other intersections are recorded on the hindsight file.

The analysis of the preceding words in this sentence generated two other, older, predictions, which are fulfilled by the alternative arguments of поолeдующих. These two intersections are noted in hindsight; but, since they actually have nothing to do with the analysis of the phrase, they will be neglected here.

The prediction pool is updated, and at the top of the pool is entered a new set of four predictions:

(1). Master (of preposition complement) (accusative plural)

(2) Compound preposition complement (accusative)

(3) Infinity

(4) End Wipe

(5) etc. (old predictions)

There are no intersections whatsoever between the single alternative argument of каскадах, /noun, locative, plural, masculine/, and the predictions in the pool. When no intersections are found during the testing of the first three predictions, the end wipe prediction is activated, and all four predictions are marked for wiping from the pool. Since the master prediction has a PSI of 01, its wiping is listed on the hindsight file. Because no intersections are found when the rest of the predictions in the pool are tested, the alternative argument is taken as the preferred argument by the arbitrary choice prediction, and the arbitrary choice attributed argument is assigned. The chain number is then incremented from 05 to 06.

As is obvious even to the casual reader of Russian, the wrong case, the accusative instead of the locative, was selected for последующих. On a following pass, this information would be sufficient to select the locative alternative argument as the preferred argument.

All the errors in prepositional phrases that have been made so far by the program occur with the preposition в, as in the preceding example. This suggests that there is an error in the ordering of the preposition complement predictions in the pool for в, since the correct prediction is always located below the one selected as the attributed argument. The priority order of the cases governed by в should be inverted, so that the locative case is tested before the accusative case. This can be done by modifying the information in the first two character positions of column 8 for the preposition.

Another example of more interesting nesting is offered by the string измерить среднюю за много периодов амплитуду (Fig. 5-6). The single alternative argument of среднюю intersects with three predictions in the pool. It fulfills the prediction of object of the verb infinitive измерить. The other two intersections with earlier object predictions are recorded on the hindsight file. A master prediction is entered at the top of the pool.

The following preposition за fulfills an infinity prediction and sets up four preposition complement predictions above the master prediction as follows:

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
TO MEASURF	V04.01	IZMERI-T,	00	31	00A-0163	VS OP30000	FO		B0B6	157 CV MAST
AVERAGE	A05.00	SREDN-JUJU	00	27	00A-0164	AD01000	--A---------	--F---------		163 OBJECT
BFHIND	I01.00	Z-A	00	29	00A-0165	R	--A-I---A-I-		IA0R00BG0680	INF PREP
MANY	D01.00	MN-OGO	00	33	00A-0166	DAXEACUNYSS	--A---------	--A---------	0T00000T0000	165 R COMP
PFRIOD	N01.00	PERIOD-OV	00	34	00A-0167	NDI1M000	-------G----	-------M----		166 R COMPM
AMPLITUDE	N04.00	AMPLITUD-U	00	33	00A-0168	NDI2F000	--A---------	--F---------		164 OBJECTM

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
TO MEASURF	V04.01	IZMERI-T,	00A-0163	VS OP30000	F-		B0B6	076540000000
AVERAGE	A05.00	SREDN-JUJU	00A-0164	AD01000	--A---------	--F---------		190910000000
BFHIND	I01.00	Z-A	00A-0165	R	--A-I---A-I-		IA0R00BG0680	057200000000
MANY	D01.00	MN-OGO	00A-0166	D XEACUNYSS	N-A---N-A---	A-A---A-A---	0T00000T0000	108195357140
PFRIOD	N01.00	PERIOD-OV	00A-0167	NDI1M000	-------G----	-------M----		142200000000
AMPLITUDE	N04.00	AMPLITUD-U	00A-0168	NDI2F000	--A---------	--F---------		002450000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT		3rd SEMI-ORGANIZED WORD	POTENTIAL ATTRIBUTED ARGUMENT
AVERAGE	A05.00	SREDN-JUJU	00A-0164	AD01000	--A---------	--F---------		156 OBJECT
AVERAGE	A05.00	SREDN-JUJU	00A-0164	AD01000	--A---------	--F---------		III L OBJ
MANY	D01.00	MN-OGO	00A-0166	DNXEACUNYSS	--A---------	--A---------	0T00000T0000	165 R COMP
MANY	D01.00	MN-OGO	00A-0166	DAXEACUNYSS	--------A---	--------A---	0T00000T0000	165 R COMP
MANY	D01.00	MN-OGO	00A-0166	DNXEACUNYSS	--------A---	--------A---	0T00000T0000	165 R COMP
MANY	D01.00	MN-OGO	00A-0166	DAXEACUNYSS	--A---------	--A---------	0T00000T0000	164 OBJECTM
MANY	D01.00	MN-OGO	00A-0166	DNXEACUNYSS	--A---------	--A---------	0T00000T0000	164 OBJECTM
MANY	D01.00	MN-OGO	00A-0166	DAXEACUNYSS	--A---------	--A---------	0T00000T0000	156 OBJECT
MANY	D01.00	MN-OGO	00A-0166	DAXEACUNYSS	--------A---	--------A---	0T00000T0000	156 OBJECT
MANY	D01.00	MN-OGO	00A-0166	DNXEACUNYSS	--A---------	--A---------	0T00000T0000	156 OBJECT
MANY	D01.00	MN-OGO	00A-0166	DNXEACUNYSS	--------A---	--------A---	0T00000T0000	156 OBJECT
MANY	D01.00	MN-OGO	00A-0166	DAXEACUNYSS	--A---------	--A---------	0T00000T0000	III L OBJ
MANY	D01.00	MN-OGO	00A-0166	DAXEACUNYSS	--------A---	--------A---	0T00000T0000	III L OBJ
MANY	D01.00	MN-OGO	00A-0166	DNXEACUNYSS	--A---------	--A---------	0T00000T0000	III L OBJ
MANY	D01.00	MN-OGO	00A-0166	DNXEACUNYSS	--------A---	--------A---	0T00000T0000	III L OBJ
AMPLITUDE	N04.00	AMPLITUD-U	00A-0168	NDI2F000	--A---------	--F---------		156 OBJECT
AMPLITUDE	N04.00	AMPLITUD-U	00A-0168	NDI2F000	--A---------	--F---------		III L OBJ

A Prepositional Phrase

Fig. 5-6

(1) Preposition complement (instrumental singular)

(2) Preposition complement (instrumental plural)

(3) Preposition complement (accusative singular)

(4) Preposition complement (accusative plural)

(5) Master (of object) (accusative singular feminine)

(6) etc. (old predictions)

The following numeral много has eight alternative arguments:

(1) /adjectival, nominative, singular/

(2) /nominal, nominative, singular/

(3) /adjectival, accusative, singular/

(4) /nominal, accusative, singular/

(5) /adjectival, nominative, plural/

(6) /nominal, nominative, plural/

(7) /adjectival, accusative, plural/

(8) /nominal, accusative, plural/

There are fourteen intersections among the alternative arguments and the predictions in the pool. The first intersection is between the third prediction in the pool and the third alternative argument, resulting in the preferred argument and attributed argument listed on the main output file. The fourth, seventh, and eighth alternative arguments also intersect with the third prediction, and there are two intersections between the fifth prediction and the third and fourth alternative arguments. These, and the remaining eight intersections are listed on the hindsight file in the order in which they are identified.

Numerals, when used adjectivally, make special master predictions dependent on the information contained in column 8: много predicts a master in the genitive case, and either singular or plural. Since the remaining unfulfilled preposition complement predictions are wiped from the pool (PSI is equal to 00), the top of the new pool after the analysis of много is as follows:

(1) Master (of preposition complement) (genitive)

(2) Compound preposition complement (accusative)

(3) Infinity

(4) End Wipe

(5) Master (of object) (accusative, singular, feminine)

(6) etc. (old predictions)

The single alternative argument of периодов intersects with the first master prediction, resulting in the attributed argument master of preposition complement. The noun preferred argument makes a new prediction of a noun complement, replacing the fulfilled master prediction at the top of the pool.

Next, the single alternative argument of амплитуду is brought into the central memory location and tested against the predictions in the pool. None of the first four predictions are fulfilled, so that the end wipe prediction is activated. This is a signal that the analysis of the prepositional phrase has been completed and the predictions of another nest are about to be tested. There is an intersection with the following master of object prediction which is noted on the main output file. Two other later intersections are then also noted. The final analysis shows the prepositional phrase за много периодов nested within the noun phrase среднюю амплитуду.

Although no mistakes have been shown in this section, it is possible that they can occur, particularly so when there is a legitimate ambiguity in the syntax that cannot be solved by syntactic analysis alone. Such a situation will be shown in the next section.

6. The Identification of the Subject, Predicate, and Object in a Clause

The recognition of the subject, predicate, and object in a clause is closely akin to the recognition of the necessary elements within any of the phrase structures. What make the subject, predicate, and object unique are the grammatical relationships among them which permit the <u>subject</u>, <u>predicate head</u>, and <u>object</u> predictions to be modified whenever one of them is fulfilled. In the existing experimental program, this is the only set of predictions that behaves in such a manner.

Whereas the subdivision of clauses into two divisions such as Chomsky's noun phrase and verb phrase[5] is the more common, in the present scheme of predictive syntactic analysis for Russian it is convenient to divide the clause into three divisions. This division adds facility to the manipulation and modification of the subroutines.

Actually four rather than three predictions are utilized to carry out the analysis of a clause, since both a <u>left object</u> prediction and an <u>object</u> prediction are used. The <u>left object</u>, which can be fulfilled by an accusative or instrumental adjective, noun, pronoun, or numeral, is predicted with the <u>subject</u> and <u>predicate head</u> predictions and must be fulfilled before the predicate has been identified, that is, it is located to the <u>left</u> of the predicate; otherwise, it is wiped from the prediction pool when the

predicate is identified and an object prediction is made based upon the verb government coding in the predicate head. Once more, it is simply a question of convenience in coding and also in the arrangement of the program output.

In a majority of cases, the subject, predicate, and object occur in the order mentioned; however, it is not uncommon to find a sentence where the positions of the subject and object are reversed. The reversed construction occurs too frequently for the analysis not to have a mechanism to recognize it. The left object prediction has been created to fulfill the need for interpreting on the first pass the sentence in which the object precedes the predicate.

There is no obvious disadvantage to this scheme of operation. Errors and mistakes due to this approach do occur, and an example of each will be considered later in this section. However, since all the alternative schemes that were considered seem to allow at least as many errors and mistakes, this approach does not seem disadvantageous.

Initially, predictions of subject, left object, and predicate head are entered into the pool in that order. The predicate head prediction is modified if either the subject or the left object predictions are fulfilled first. Likewise, the predicate head prediction modifies the subject prediction when the predicate head is fulfilled first. The modifications serve to limit the number of alternative arguments that can intersect with the modified predictions. This is particularly important because of the frequency of occurrence of nouns and adjectives with at least two alternative arguments, one nominative and the other either accusative or instrumental. Frink and Kline[6] have compiled some statistics on the frequency of the textual occurrence

of the various alternative arguments. Some of the figures based on a sample of 9,618 nouns and adjectives found in texts are given in Table 5-4. It is seen that more than one third of all nouns and adjectives have nominative-accusative or nominative-instrumental alternative argument pairs which can fulfill both <u>subject</u> and <u>object</u> (or <u>left object</u>) predictions. Without the modifications of predictions for agreement in case and number, errors in analysis would occur more often, and more passes would be needed to achieve a correct analysis.

	Nouns		Adjectives		Nouns and Adjectives	
Words with alternative arguments that can fulfill both <u>subject</u> and <u>object</u> predictions.	2,706	44.0%	878	25.3%	3,584	37.3%
Words with alternative arguments that can fulfill either <u>subject</u> or <u>object</u> predictions.	938	15.2%	2,429	70.1%	3,367	35.0%
Words with alternative arguments that can fulfill neither <u>subject</u> nor <u>object</u> predictions.	2,509	40.8%	158	4.6%	2,667	27.7%
	6,153		3,465		9,618	

Frequency with which Text Occurrences of Nouns and Adjectives Can Fulfill <u>Subject</u> and <u>Object</u> Predictions

TABLE 5-4

To illustrate the effect of prediction modification, several examples will be used, and predictions which do not affect the modifications of interest will be deliberately overlooked.

The most common sequence is subject-predicate-object, which is represented by the sentence segment здесь мы определим значения... (Fig. 5-7). The subject, left object, and predicate head predictions are in the pool in the given order. Placing the subject above the left object permits a word whose alternative arguments intersect with both predictions to be selected as the subject.

The first word, the adverb здесь, is accepted by the infinity prediction. Since an adverb makes no new predictions the pool remains unmodified. The pronoun мы has only one alternative argument, /pronoun, nominal, first person, nominative, plural, masculine or feminine/, which intersects only with the subject prediction. Since there are no other intersections, nothing is recorded on the hindsight file. In updating the prediction pool, the predicate head prediction is modified so that only a first person, plural, and masculine or feminine predicate can fulfill the prediction. The following verb определим satisfies these imposed conditions so that it can be selected as the predicate head. The second alternative argument of определим, /short form adjective, singular, masculine/, cannot satisfy the conditions imposed on the predicate head prediction since the alternative argument is singular and the modification is for plural only.

Since the predicate head has been fulfilled before the left object, the latter prediction is wiped from the pool and a prediction for an accusative object, based on the "P7" government code in column 5 of определим,

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT	3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
HERE	I01.00	ZD-ES.	00	20	00A-1966	H			INF ADVB
WE	P01.00	M-Y	00	20	00A-1967	PN A PVP 0	-------N------ -------H-------		III SUBJCT
WILL FIND	V04.00	OPREDEL-IM	00	11	00A-1968$	VS00P70000	000V00CAD0	B2B4	III V PRED
SIGNIFICANCE	N10.00	ZNACHENI-JA	00	08	00A-1969	NDI1N000	---------A--- ---------N---		968 OBJECT

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
HERE	I01.00	ZD-ES.	00A-1966	H			072080000000
WF	P01.00	M-Y	00A-1967	PN A PVP 0	-------N------ -------H------		102313333338
DFFINABLE	A05.00	OPREDEL-IM	00A-19681	AD00000 1 1	N------------- M-------------		127650000000
WILL FIND	V04.00	OPREDEL-IM	00A-1968$	VS00P70000	---V--CAD-	B2B4	127660000000
SIGNIFICANCE	N10.00	ZNACHENI-JA	00A-1969	NDI1N000	-G----N-A--- -N----N-N---		072780000000

A Segment of a Clause

Fig. 5-7

is entered into the new pool. Although значения has three alternative arguments, there is only one intersection and the noun is selected as the object of определим.

An example of an adjective predicate head preceding a subject is given in the next illustration, предложена методика...(Fig. 5-8). The single alternative argument of предложена intersects with the predicate head prediction. The subject prediction is then modified so that only a singular and feminine subject can be accepted. Since методика fulfills these limitations, it is accepted as the subject of the clause.

As an example of how the modification of predictions catches errors, consider the string при подключении следующего накапливающего конденсатора все явления повторяются...(Fig. 5-9). The subject, left object, and predicate head predictions are in the pool together with an infinity prediction. The preposition при is accepted by the infinity prediction which leads to the identification of the prepositional phrase при подключении следующего накапливающего конденсатора (Sec. 5). After the analysis of конденсатора, a prediction for a noun complement is placed above the other predictions of the clause.

The pronoun все has eight alternative arguments:

(1) /pronoun, adjectival, nominative, singular, neuter/

(2) /pronoun, adjectival, accusative, singular, neuter/

(3) /pronoun, adjectival, nominative, plural/

(4) /pronoun, adjectival, accusative, plural/

(5) /pronoun, nominal, nominative, singular, neuter/

(6) /pronoun, nominal, accusative, singular, neuter/

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT	3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
OFFERED	A01.00	PREDLOZHEN-A	00	20	00A-0021	AD0000 13	N----------- F-----------		III A PRED
METHOD	N04.10	METODIK-A	00	08	00A-0022	NDI1F000	N----------- F-----------		III SUBJCT

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	DICTIONARY SERIAL NO.
OFFERFD	A01.00	PREDLOZHEN-A	00A-0021	AD0000 13	N----------- F-----------	154960000000
METHOD	N04.10	METODIK-A	00A-0022	NDI1F000	N----------- F-----------	106670000000

A Segment of a Clause

Fig. 5-8

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT		CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO	SIZE OF TEXT	TEXT SERIAL NO	ORGANIZED WORD		PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT	
IN THE TIME	OF	I01.00	PR-I	00	20	00A-2143	R		------P-----P		POOR00A00600	INF	PREP
CONNECTION		N10.00	PODKLJUCHENI -I	00	22	00A-2144	NDI1N000		------P-------	------N-------		143	R COMP
FOLLOWING		A04.00	SLEDUJUSHCH- EGO	00	25	00A-2145	AD0100	4	-G------------	-B------------		144	N COMP
STORING		A04.00	NAKAPLIVAJUS HCH-EGO	00	27	00A-2146	AD01000	4	-G------------	-B------------		145	N COMPM
CAPACITOR		N01.00	KONDENSATOR- A	00	27	00A-2147	NDI1M000		-G------------	-M------------		146	N COMPM
ALL		P01.00	VS-E	00	27	00A-2148	PA K ATF	0	N-------------	N------------		III	SUBJCT
APPEARANCE		N10.00	JAVLENI-JA	00	11	00A-2149	NDI1N000		---------A---	---------N---		III	L OBJ
TO REPEAT		V01.00	POVTORJA-JUT SJA	01	10	00A-2150	VNR0000000		00000TBADR		B0B1B4B6	III	ARBTR

TEXTHADIC

FIRST ENGLISH EQUIVALENT		CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO	ORGANIZED WORD		CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO
IN THE TIME	OF	I01.00	PR-I	00A-2143	R		-----P-----P		POOR00A00600	15424555555[illegible]
CONNECTION		N10.00	PODKLJUCHENI -I	00A-2144	NDI1N000		-----P------	-----N------		14709500000C
FOLLOWING		A04.00	SLEDUJUSHCH- EGO	00A-2145	AD0100	4	-GA----------	-BM---------		185380000000
STORING		A04.00	NAKAPLIVAJUS HCH-EGO	00A-2146	AD01000	4	-GA----------	-BM---------		112850000000
CAPACITOR		N01.00	KONDENSATOR- A	00A-2147	NDI1M000		-G-----------	-M----------		091240000000
ALL		P01.00	VS-E	00A-2148	PK K ATF	0	N-A---N-A---	N-N---A-A---		027255000000
APPEARANCE		N10.00	JAVLENI-JA	00A-2149	NDI1N000		-G----N-A---	-N----N-N---		219260000000
TO REPEAT		V01.00	POVTORJA-JUT SJA	00A-2150	VNR0000000		-----TBADR		B0B1B4B6	145130000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT		CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO	ORGANIZED WORD		ALTERNATIVE ARGUMENT		POTENTIAL ATTRIBUTED ARGUMENT	
FOLLOWING		A04.00	SLEDUJUSHCH- EGO	00A-2145	AD0100	4	-GA----------	-BM----------	III	FRASER
FOLLOWING		A04.00	SLEDUJUSHCH- EGO	00A-2145	AD0100	4	--A----------	--M----------	III	L OBJ
STORING		A04.00	NAKAPLIVAJUS HCH-EGO	00A-2146	AD01000	4	-GA----------	-BM----------	III	FRASER
STORING		A04.00	NAKAPLIVAJUS HCH-EGO	00A-2146	AD01000	4	--A----------	--M----------	III	L OBJ
ALL		P01.00	VS-E	00A-2148	PA K ATF	0	------N------	-------A-----	III	SUBJCT
ALL		P01.00	VS-E	00A-2148	PN K ATF	0	N------------	N------------	III	SUBJCT
ALL		P01.00	VS-E	00A-2148	PN K ATF	0	------N------	-------A-----	III	SUBJCT
ALL		P01.00	VS-E	00A-2148	PA K ATF	0	--A----------	--N----------	III	L OBJ
ALL		P01.00	VS-E	00A-2148	PA K ATF	0	---------A---	---------A---	III	L OBJ
ALL		P01.00	VS-E	00A-2148	PN K ATF	0	--A----------	--N----------	III	L OBJ
ALL		P01.00	VS-E	00A-2148	PN K ATF	0	---------A---	---------A---	III	L OBJ
PREDICTION	WIPED	----	148013000972							
PREDICTION	WIPED	----	III012001915							

A Segment of a Clause

Fig. 5-9

(7) /pronoun, nominal, nominative, plural/

(8) /pronoun, nominal, accusative, plural/

Four of these alternative arguments intersect with the subject prediction, and the other four intersect with the left object prediction. Although все is correctly identified as the subject of the clause, the wrong preferred argument is selected, /pronoun, adjectival, 3rd person, nominative, singular, neuter/. All of the seven other intersections are recorded on the hindsight file. The subject attributed argument modifies the predicate head prediction, so that only a singular and neuter predicate can fulfill the prediction. The adjectival preferred argument sets up a master prediction, which must be fulfilled (PSI 01), followed by an end wipe. A noun or adjective fulfilling the master prediction must be nominative, singular, and neuter.

When the three alternative arguments of явления are brought into the central memory location, none of them intersects with the leading master prediction. The end wipe is activated, wiping the master prediction and noting it on the hindsight file. The /noun, accusative, plural, neuter/ alternative argument intersects with the left object prediction. This further modifies the predicate head prediction so that only a transitive verb can be accepted.

The verb повторяются cannot fulfill the predicate head prediction, since it is plural and reflexive. It cannot fulfill any other prediction either, so that it is selected as an arbitrary choice after the predicate head prediction is wiped and recorded on the hindsight file. The chain number is incremented to indicate the error. In a later pass, if the subject prediction were initially limited to plural subjects, the analysis would proceed correctly.

The sentence: Предметом настоящего сообщения является анализ возможностей... (Fig. 5-10), is an example of the sequence object-predicate-subject, which is quite common when there is a reflexive verb acting as the predicate. The subject, left object, and predicate head predictions are at the top of the pool when the alternative argument of предметом, /noun, instrumental, singular, masculine/, is tested. The single intersection with the left object prediction results in the selection of предметом as the object of the clause. The predicate head is modified so that only a predicate governing the instrumental case can be accepted. The noun phrase настоящего сообщения is selected as the noun complement of предметом (see Sec. 5), after which the alternative argument of является is tested. Once more there is a single intersection, this time with the modified predicate head prediction. The program can determine that the verb governs the instrumental case by testing whether the verb is reflexive. Having selected a predicate before finding a subject, the subject prediction is modified so that only a third person singular subject can be accepted. Although there are two alternative arguments of анализ, there is only one intersection, and анализ is chosen as the subject of the clause.

Another sentence, В эту емкость помимо распределенной емкости монтажа входят междуэлектродные емкости всех подключающих ламп. (Fig. 5-11), demonstrates that the ordering of the predictions can occasionally cause an error. The sentence starts with two prepositional phrases в эту емкость and помимо распределенной емкости монтажа. The next word, the verb входят, fulfills the predicate head prediction, on the one hand, modifying the subject prediction so that only a third person plural subject will be accepted and, on the other hand, after wiping the left object prediction, introducing an

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT	3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
OPJECT	N01.00	PREDMET-OM	00	20	00A-0428	NDI1M000	----I-------- -----M--------		III L OBJ
PRESENT	A04.00	NASTOJASHCH- EGO	00	11	00A-0429	AD0000C	-G----------- -B-----------		428 N COMP
COMMUNICATIO N	N10.00	SOOPSHCHENI- JA	00	14	00A-0430	NDI1N000	-G----------- -N-----------		429 N COMPM
TO BE	V01.00	JAVLJA-FTSJA	00	14	00A-0431	VN 0000000	00T000BADR	B0B1B4B6	III V PRED
ANALYSIS	N01.00	ANALIZ-	00	09	00A-0432	NDI1M000	N------------ M------------		III SUBJCT
POSSIPILITY	N06.00	VOZMOZHNOST- EJ	00	11	00A-0433	NDI1F000	--------G---- --------F----		432 N COMP

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
OPJECT	N01.00	PREDMET-OM	00A-0428	NDI1M000	----I-------- ----M--------		155030000000
PRESENT	A04.00	NASTOJASHCH- EGO	00A-0429	AD00000	-GA---------- -BM----------		115260000000
COMMUNICATIO N	N10.00	SOOBSHCHENI- JA	00A-0430	NDI1N000	-G----N-A---- -N----N-N----		187950000000
TO BE	V01.00	JAVLJA-ETSJA	00A-0431	VN 0000000	--T---BADR	B0B1B4B6	219270000000
ANALYSIS	N01.00	ANALIZ-	00A-0432	NDI1M000	N-A---------- M-M----------		002500000000
POSSIPILITY	N06.00	VOZMOZHNOST- EJ	00A-0433	NDI1F000	--------G---- --------F----		021500000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT	POTENTIAL ATTRIBUTED ARGUMENT
COMMUNICATIO N	N10.00	SOOBSHCHENI- JA	00A-0430	NDI1N000	------N------ -------N------	III SUBJCT

A Segment of a Clause

Fig. 5-10

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD		PREFERRED ARGUMENT	3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
IN(TO)	I01.00	-V	00	20	00A-2013	R		--A--P--A--P	APOROOBA0650	INF PREP
THIS	P01.00	EHT-U	00	24	00A-2014	PA K STD	0	--A---------- --F----------		013 R COMP
CAPACITY	N06.00	EMKOST-,	00	24	00A-2015	NDI1F000		--A---------- --F----------		014 R COMPM
BESIDES	I01.00	POMIM-O	00	24	00A-2016	RH		-G-----G----	GOORHO100200	INF PREP
DISTRIBUTED	A01.00	RASPREDELENN -OJ	00	25	00A-2017	AD0000	3	-G----------- -F----------		016 R COMP
CAPACITY	N06.00	EMKOST-I	00	27	00A-2018	NDI1F000		-G----------- -F----------		017 R COMPM
ASSEMBLING	N03.10	MONTAZH-A	00	27	00A-2019	NDI2M000		-G----------- -M----------		018 N COMP
ENTER	V04.00	VXOD-JAT	00	30	00A-2020	VN 0C40000		00000TBADO	B1B4B5	III V PRED
INTERELECTRO DE	A02.00	MEZHDUEHLEKT RODN-YE	00	09	00A-2021	AD00000		---------A--- ---------A---		020 OBJECT
CAPACITY	N06.00	EMKOST-I	00	11	00A-2022	NDI1F000		---------A--- ---------F---		021 OBJECTM
ALL	P01.00	VSEX-	00	11	00A-2023	PA K PTF	0	--------G---- --------A----		022 N COMP
CUTTING IN	A04.00	PODKLJUCHAJU SHCH-IX	00	14	00A-2024	AD0100	4	--------G---- --------A----		023 N COMPM
TUBE	N04.00	LAMP-	00	14	00A-2025	NDI2F000		--------G---- --------F----		024 N COMPM
*.		*.	00	14	00A-2026	.	0			END OF SENT.
					TEXTHADIC					
								CODING DUE TO WORD-BY-WORD ANALYSIS		DICTIONARY SERIAL NO.
IN(TO)	I01.00	-V			00A-2013	R		--A--P--A--P	APOROOBA0650	000020000000
THIS	P01.00	EHT-U			00A-2014	PK K STD	0	--A---------- --F----------		218928958326
CAPACITY	N06.00	EMKOST-,			00A-2015	NDI1F000		N-A---------- F-F----------		056460000000
BESIDES	I01.00	POMIM-O			00A-2016	RH		-G-----G----	GOORHO100200	151700000000
DISTRIBUTED	A01.00	RASPREDELENN -OJ			00A-2017	AD0000	3	-G-CIP------- -F-FFF------		173300000000
CAPACITY	N06.00	EMKOST-I			00A-2018	NDI1F000		-G-C-PN-A---- -F-F-FF-F----		056460000000
ASSEMBLING	N03.10	MONTAZH-A			00A-2019	NDI2M000		-G----------- -M----------		110280000000
ENTER	V04.00	VXOD-JAT			00A-2020	VN 0C40000		-----TBAD-	B1B4B5	030000000000
INTERELECTRO DE	A02.00	MEZHDUEHLEKT RODN-YE			00A-2021	AD00000		-------N-A--- -------A-A---		105360000000
CAPACITY	N06.00	EMKOST-I			00A-2022	NDI1F000		-G-C-PN-A---- -F-F-FF-F----		056460000000
ALL	P01.00	VSEX-			00A-2023	PK K PTF	0	--------GA--P --------AA--A		027685000000
CUTTING IN	A04.00	PODKLJUCHAJU SHCH-IX			00A-2024	AD0100	4	--------GA--P --------AA--A		147080000000
TUBE	N04.00	LAMP-			00A-2025	NDI2F000		--------G---- --------F----		098680000000
*.		*.			00A-2026		*			
					HINDSIGHT					
								ALTERNATIVE ARGUMENT		POTENTIAL ATTRIBUTED ARGUMENT
THIS	P01.00	EHT-U			00A-2014	PN K STD	0	--A---------- --F----------		013 R COMP
THIS	P01.00	EHT-U			00A-2014	PA K STD	0	--A---------- --F----------		III L OBJ
THIS	P01.00	EHT-U			00A-2014	PN K STD	0	--A---------- --F----------		III L OBJ
CAPACITY	N06.00	EMKOST-,			00A-2015	NDI1F000		N------------ F-----------		III SUBJCT
CAPACITY	N06.00	EMKOST-,			00A-2015	NDI1F000		--A---------- --F----------		III L OBJ
DISTRIBUTED	A01.00	RASPREDELENN -OJ			00A-2017	AD0000	3	-G-CIP------- -F-FFF------		III FRASER
DISTRIBUTED	A01.00	RASPREDELENN -OJ			00A-2017	AD0000	3	----I-------- ----F-------		III L OBJ
CAPACITY	N06.00	EMKOST-I			00A-2018	NDI1F000		-------N----- -------F-----		III SUBJCT
CAPACITY	N06.00	EMKOST-I			00A-2018	NDI1F000		---------A--- ---------F---		III L OBJ
INTERELECTRO DE	A02.00	MEZHDUEHLEKT RODN-YE			00A-2021	AD00000		-------N----- -------A-----		III SUBJCT
CAPACITY	N06.00	EMKOST-I			00A-2022	NDI1F000		-------N----- -------F-----		III SUBJCT
ALL	P01.00	VSEX-			00A-2023	PN K PTF	0	--------G---- --------A----		022 N COMP
PREDICTION WIPED	----	III012003900								
*.		*.			00A-2026	.	0			END OF SENT.

A Complete Sentence

Fig. 5-11

accusative <u>object</u> prediction in the pool above the <u>subject</u> prediction. The alternative arguments of the following adjective, междуэлектродные, intersect with both the <u>object</u> and <u>subject</u> predictions. The attributed argument is <u>object</u> since the first intersection is with the <u>object</u> prediction. Емкости fulfills the <u>master</u> prediction generated by the preferred argument of междуэлектродные, and всех подключающих ламп is a <u>noun complement</u> noun structure predicted by емкости. When the period, the punctuation mark indicating the end of the sentence, is identified, all remaining predictions that should have been fulfilled are recorded on the hindsight file. In this case, the only such prediction is the unfulfilled <u>subject</u> prediction. This would allow a future pass to identify междуэлектродные correctly as the subject of the clause.

This particular error can be attributed to the relatively unsophisticated way of handling <u>object</u> predictions. In the experimental program verbs are assumed to govern accusative objects, unless either the verb is in the reflexive voice, or there is a government code in column 5 for another case. A future program should be capable of interpreting phrase government in addition to object government.

A sentence which cannot be analyzed uniquely by syntactic analysis alone is illustrated by: В локационной технике большое распространение получило применение ...(Fig. 5-12). The noun phrase большое распространение and the noun применение both have the same two alternative arguments, /nominative, singular, neuter/ and /accusative, singular, neuter/. On a first pass the noun phrase preceding the verb will be selected as the subject, while the noun phrase following the verb will be selected as the object, and an indication will be made on the hindsight file that the first noun phrase

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
IN(TO)	I01.00	-V	00	20	00A-0213	R	--A--P--A--P		APOROOBA0650	INF PREP
RADIO LOCATI ON	A01.00	LOKATSIONN-O J	00	24	00A-0214	AD00000	-----P------	-----F------		213 R COMP
TECHNOLOGY	N04.10	TEXNIK-F	00	24	00A-0215%	NDI1F100	-----P------	-----F------		214 R COMPM
BIG	A08.00	BOL.SH-OE	00	24	00A-0216	AD01000	N-----------	N-----------		III SUBJCT
PROPAGATION	N16.00	RASPROSTRANF NI-E	00	11	00A-0217	NDI1N100	N-----------	N-----------		216 SUBJCTM
RECEIVED	V04.20	POLUCHIL-O	00	11	00A-0218	VS00P70000	SSS000AND0		B3 T4	III V PRED
APPLICATION	N10.00	PRIMENENI-E	00	08	00A-0219	NDI1N000	--A---------	--N---------		218 OBJECT

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO
IN(TO)	I01.00	-V	00A-0213	R	--A--P--A--P		APOROOBA0650	000020000000
RADIO LOCATI ON	A01.00	LOKATSIONN-O J	00A-0214	AD00000	-G-CIP------	-F-FFF------		101196666666
TECHNICIAN	N01.10	TEXNIK-F	00A-02151	NDA1M000	-----P------	-----M------		197710000000
TECHNOLOGY	N04.10	TEXNIK-F	00A-0215%	NDI1F100	---C-P------	---F-F------		197720000000
BIG	A08.00	BOL.SH-OE	00A-0216	AD01000	N-A---------	N-N---------		009100000000
PROPAGATION	N16.00	RASPROSTRANF NI-E	00A-0217	NDI1N100	N-A---------	N-N---------		173480000000
RECEIVED	V04.20	POLUCHIL-O	00A-0218	VS00P70000	SSS---AND-		B3 T4	151170000000
APPLICATION	N10.00	PRIMENENI-E	00A-0219	NDI1N000	N-A---------	N-N---------		159010000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT		POTENTIAL ATTRIBUTED ARGUMENT
RADIO LOCATI ON	A01.00	LOKATSIONN-O J	00A-0214	AD00000	----I-------	----F-------	III L OBJ
BIG	A08.00	BOL.SH-OE	00A-0216	AD01000	--A---------	--N---------	III L OBJ
PROPAGATION	N16.00	RASPROSTRANF NI-E	00A-0217	NDI1N100	--A---------	--N---------	III L OBJ

A Segment of a Clause

Fig. 5-12

might have been the object. It is obvious that if the sentence had read ...применение получило большое распространение..., the analyzed output would be a syntactic analysis on the first pass which a reader of Russian would immediately reject on semantic grounds, but which the experimental program would accept as a correct syntactic analysis. This is an example of a mistake in the program as opposed to an error.

7. Comma

Nested structures can be written in several notations in artificial languages. Oettinger has demonstrated four of these notations:[7] left-parenthetic, right-parenthetic, full parenthetic, and parenthesis-free. Similar notations are used in the Russian language. A structure similar to the left-parenthetic notation is the prepositional phrase, where the preposition serves as an implicit left parenthesis. Likewise, the initial adjective of a noun structure can be considered an implicit left parenthesis.

A notation equivalent to the full-parenthetic notation is also used in the Russian language. The most trivial example is the explicit use of the left- and right-parentheses to isolate a side comment within a paragraph or even within an individual sentence. Nested structures such as participial phrases and clauses are isolated from the rest of a sentence by commas. Here, since only one symbol is used, a "left-comma" cannot be distinguished from a "right-comma".

In the experimental program, the comma is recognized only in its function of a phrase or clause separator. Other uses of the comma such as separating words or phrases used in series have not been studied yet.

A comma may occur at any point in a sentence following the initial word. Generally, there is no signal that the comma is about to occur. It is necessary, therefore, to accept one of these punctuation marks with the infinity prediction, and then to make a set of predictions of all the structures that the comma might precede. Presently, three predictions are used for this purpose: the phraser prediction that predicts gerunds and participles, the relative conjunction prediction that predicts conjunctions which introduce subordinate clauses, and the relative pronoun prediction that predicts relative pronouns which also introduce subordinate clauses. A relative conjunction such as когда has no syntactic role within the clause that it introduces, whereas a relative pronoun such as который serves as a noun or adjective within the clause that it introduces. Prepositional phrases which might be offset by commas from the rest of the sentence are also predicted, since there is an infinity prediction present in this set also. Twelve predictions are entered at the top of the prediction pool after the identification of a comma as follows:

(1) Phraser

(2) Infinity

(3) End Wipe

(4) Relative Conjunction

(5) Infinity

(6) End Wipe

(7) Relative Pronoun

(8) Subject (Inactive)

(9) Left Object (Inactive)

(10) Predicate Head (Inactive)

(11) Infinity

(12) End Wipe

The inactive subject, predicate head, and left object predictions are not tested. Only when they are activated, that is, their PSI is reduced by 50 (see Appendix F), are they tested. These predictions are activated only after the fulfillment of either the relative conjunction or the relative pronoun predictions. They then serve as the predictions of the clause introduced by the relative conjunction or relative pronoun.

If only the use of a comma to isolate nested structures is considered, a comma can serve two functions. On the one hand, it is used to introduce a new nested structure and, on the other hand, it is used to indicate the end of a nested structure and the return to a preceding nested structure which had not been finished. This is illustrated schematically in Fig. 5-13. The commas have been numbered for identification. Comma-1 introduces a new nested structure, the second clause. Likewise, comma-2 introduces a new nested structure, the participial phrase. Both comma-3 and comma-4 indicate the end of a nested structure and the return to a previously uncompleted structure, comma-3 to clause 2 and comma-4 to clause 1.

	(1)		(2)		(3)		(4)	
________	,	________	,	________	,	________	,	________.
Clause 1		Clause 2		Participial Phrase		Clause 2		Clause 1

Schematic Representation of Nested Structures in a Sentence

Fig. 5-13

In the event a comma is being used to indicate return to an uncompleted nested structure, none of the predictions, phraser, relative conjunction, or relative pronoun, should be fulfilled. If an end wipe prediction is placed below the set of predictions made by a comma, all of these unfulfilled predictions should be wiped from the pool. Infinity and end wipe predictions are placed underneath each of the three introductory predictions, so that if a relative conjunction prediction is fulfilled, the phraser prediction is immediately wiped from the pool, and if a relative pronoun prediction is fulfilled, both the phraser and relative conjunction predictions are wiped. The ordering of the phraser, relative conjunction, and relative pronoun predictions is based on the possibility of multiple intersections between these predictions and the alternative arguments of a word, and the desirable initial guess of the preferred argument.

The inactive subject, left object, and predicate head predictions are put into the pool at the same time as the relative conjunction and the relative pronoun, so that they may be at their proper level of nesting when a subordinate clause is positively identified. This will be made more obvious by several examples.

As an example of the identification of a participial phrase, consider the sentence: Нелинейные искажения в элементах схемы, осуществляющих усреднение, неизбежно приведут... (Fig. 5-14). Нелинейные искажения is identified as the subject noun phrase, after which the prepositional phrase в элементах схемы is identified. The following comma is accepted by the infinity prediction, and the phraser, relative conjunction, and relative pronoun predictions are inserted into the pool above the original unfulfilled

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
NONLINEAR	A02.00	NELINEJN-YE	00	20	00A-1255	AD00000	-------N------	-------A------		III SUBJCT
DISTORTION	N10.00	ISKAZHEN'-JA	00	11	00A-1256	NDI1N000	-------N------	-------N------		255 SUBJCTM
IN(TO)	I01.00	-V	00	11	00A-1257	R	--A--P--A--P		APOROOBA0650	INF PREP
ELEMENT	N01.00	EHLEMENT-AX	00	14	00A-1258	NDI1M000	------------P	------------M		257 R COMP
CIRCUIT	N04.00	SXEM-Y	00	14	00A-1259	NDI2F0Y0	-G-----------@	-F-----------		258 N COMP
*,		*,	00	17	00A-1260	• O				INF COMMA
CARRYING OUT	A04.00	OSUSHCHESTVL JAJUSHCH-I	00	28	00A-1261	AD0100 4	---------GA--P	---------AA--A		260 FRASER
AVERAGING	N10.00	USREDNENI-E	00	31	00A-1262	NDI1N000	--A-----------	--N-----------		261 OBJECT
*,		*,	00	33	00A-1263	• O				INF COMMA
INEVITABLE	A02.00	NEIZBEZHN-O	00	39	00A-1264	AD00000 2	N------------	N------------		INF ADVB
TO HAPPEN	V08.20	PRIVED-UT	00	39	00A-12651	VS KP20000	00000TCAD0		B2B4	III V PRED

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS			3rd SEMI-ORGANIZED WORD		DICTIONARY SERIAL NO.
NONLINEAR	A02.00	NELINEJN-YE	00A-1255	AD00000	-------N-A----	-------A-A----				117280000000
DISTORTION	N10.00	ISKAZHENI-JA	00A-1256	NDI1N000	-G----N-A----	-N-----N-N----				081560000000
IN(TO)	I01.00	-V	00A-1257	R	--A--P--A--P			APOROOBA0650		000020000000
ELEMENT	N01.00	EHLEMENT-AX	00A-1258	NDI1M000	------------P	------------M				218360000000
CIRCUIT	N04.00	SXEM-Y	00A-1259	NDI2F0Y0	-G----N-A----	-F-----F-F----				194360000000
*,		*,	00A-1260	*						
CARRYING OUT	A04.00	OSUSHCHESTVL JAJUSHCH-IX	00A-1261	AD0100 4	---------GA--P	---------AA--A				130450000000
AVERAGING	N10.00	USREDNENI-E	00A-1262	NDI1N000	N-A-----------	N-N-----------				206195000000
*,		*,	00A-1263	*						
INEVITABLE	A02.00	NEIZBEZHN-O	00A-1264	AD00000 2	N------------	N------------				116800000000
TO HAPPEN	V08.20	PRIVED-UT	00A-12651	VS KP20000	------TCAD-			B2B4		157825000000
WILL BRING	V08.20	PRIVED-UT	00A-1265$	VS00P300K0	------TCAD-		P309	B2B4B5	T4	157790000000

HIND SIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT			3rd SEMI-ORGANIZED WORD		POTENTIAL ATTRIBUTED ARGUMENT
NONLINEAR	A02.00	NELINEJN-YE	00A-1255	AD00000	----------A----	----------A----				III L OBJ
DISTORTION	N10.00	ISKAZHENI-JA	00A-1256	NDI1N000	----------A----	----------N----				III L OBJ
CIRCUIT	N04.00	SXEM-Y	00A-1259	NDI2F0Y0	----------A----	----------F----				III L OBJ
CARRYING OUT	A04.00	OSUSHCHESTVL JAJUSHCH-IX	00A-1261	AD0100 4	----------A----	----------A----				III L OBJ
AVERAGING	N10.00	USREDNENI-E	00A-1262	NDI1N000	--A-----------	--N-----------				III L OBJ
WILL BRING	V08.20	PRIVED-UT	00A-1265$	VS00P300K0	00000TCAD0		P309	B2B4B5	T4	III V PRED

A Segment of a Sentence with a Participial Phrase

Fig. 5-14

predicate head and left object predictions. The two sets of predictions are separated by an end wipe prediction. Осуществляющих fulfills the phraser prediction and is identified as a participle.

To digress somewhat at this point, it should be pointed out that осуществляющих is tested for being a participle by the phraser prediction (the first intersection) and is tested for being an adjective by the left object prediction (the second intersection). Although the two tests are performed on the same word, they are entirely independent, the phraser prediction not recognizing that the word might be an adjective, and vice versa.

After the identification of the participle, a prediction for an object of the participle is made and fulfilled by the following word, the noun усреднение. The following comma makes a new set of predictions of phraser, relative conjunction, and relative pronoun. A schematic diagram of the prediction pool at this point is given in Fig. 5-15. Three nested structures are evident at this time. At the top of the pool are the predictions referring to a yet unidentified possible third nested structure. Below are the predictions generated during the analysis of the already identified participial phrase as well as the residue of unfulfilled predictions due to the first comma. At the bottom of the pool are the original predictions from the main clause which have not been unfulfilled yet.

The first word after the second comma, неизбежно, is identified as an adverb by the infinity prediction. Adverbs, like prepositions, often cannot be predicted and must be satisfied by the infinity prediction. In the present experimental system, the wiping of the prediction pool is

(1) Predictions by 2nd Comma

End Wipe

(2) Predictions by Participial Phrase and 1st comma

End Wipe

(3) Main Clause Unfulfilled Predictions

Schematic Diagram of Prediction Pool After Analysis
of Word 00A-1263 (Fig. 5-14)

Fig. 5-15

inhibited when an adverb is identified, so that the prediction pool after the analysis of неизбежно is identical to the pool before the analysis of the adverb. The two alternative arguments of приведут are then brought into the central memory location. These two alternative arguments are almost identical on a syntactic level. The only difference is that the first alternative argument governs the dative case, indicated by the "P2", while the second alternative argument governs the accusative case, indicated by the "P3" in column 5.

Neither of the alternative arguments intersects with any of the predictions made by the second comma. This indicates that a new nested

structure has not been found. Continuing the testing, no intersections are found with the predictions from the participial phrase or from the first comma. Wiping this second set of predictions leaves only the predictions from the main clause. The predicate head prediction intersects with both alternative arguments and, as usual, the first is selected as the preferred argument. The intersection shows that the sentence has indeed reverted back to the main clause.

The analysis of a subordinate clause introduced by a relative conjunction or a relative pronoun is not as straightforward as the analysis of a participial phrase, chiefly because it is necessary to consider the subject-predicate-object structure within the clause. A series of illustrations will make the difficulties clear.

Consider first the subordinate clause: ...,который позволяет изучать сигналы... (Fig. 5-16). The relative pronoun который has four alternative arguments:

(1) /relative pronoun, adjectival, nominative, 3rd person, singular/

(2) /relative pronoun, adjectival, accusative, 3rd person, singular/

(3) /relative pronoun, nominal, nominative, 3rd person, singular/

(4) /relative pronoun, nominal, accusative, 3rd person, singular/

The first intersection between an alternative argument and one of the comma predictions is between the relative pronoun prediction and the first alternative argument. When the relative pronoun prediction is fulfilled, the testing process is temporarily suspended. A special subroutine scans the prediction pool, activating any inactive predictions in the pool, in this case a subject,

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT		CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT	3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
*.			*.	00	20	00A-0810		O		INF COMMA
WHICH		P01.00	KOTOR-YJ	00	31	00A-0811	PA K STRITO	N----------- M-----------		810 SUBJCT
TO ALLOW		V01.00	POZVOLJA-ET	00	27	00A-0812	VN OP20000	00T000BADO	B0B1B4B6	810 V PRED
TO STUDY		V01.00	IZUCHA-T,	00	24	00A-0813	VN OP30000	FO	B0B1B4B6	812 V MAST
SIGNAL		N01.00	SIGNAL-Y	00	26	00A-0814	NDI1M000	--------A--- ---------M---		813 OBJECT
						TEXTHADIC		**CODING DUE TO WORD-BY-WORD ANALYSIS**		**DICTIONARY SERIAL NO.**
*.			*.			00A-0810		*		
WHICH		P01.00	KOTOR-YJ			00A-0811	PK K STRITO	N-A--------- M-M---------		095121632647
TO ALLOW		V01.00	POZVOLJA-ET			00A-0812	VN OP20000	--T---BAD-	B0B1B4B6	149050000000
TO STUDY		V01.00	IZUCHA-T,			00A-0813	VN OP30000	F-	B0B1B4B6	078090000000
SIGNAL		N01.00	SIGNAL-Y			00A-0814	NDI1M000	------N-A--- ------M-M---		183370000000
						HINDSIGHT		**ALTERNATIVE ARGUMENT**		**POTENTIAL ATTRIBUTED ARGUMENT**
WHICH		P01.00	KOTOR-YJ			00A-0811	PN K STRITO	N----------- M-----------		810 SUBJCT
WHICH		P01.00	KOTOR-YJ			00A-0811	PA K STRITO	--A--------- --M---------		810 L OBJ
WHICH		P01.00	KOTOR-YJ			00A-0811	PN K STRITO	--A--------- --M---------		810 L OBJ
WHICH		P01.00	KOTOR-YJ			00A-0811	PA K STRITO	--A--------- --M---------		803 OBJECT
WHICH		P01.00	KOTOR-YJ			00A-0811	PN K STRITO	--A--------- --M---------		803 OBJECT
WHICH		P01.00	KOTOR-YJ			00A-0811	PA K STRITO	N----------- M-----------		III SUBJCT
WHICH		P01.00	KOTOR-YJ			00A-0811	PN K STRITO	N----------- M-----------		III SUBJCT
PREDICTION	WIPED	----	811013000972							
TO STUDY		V01.00	IZUCHA-T			00A-0813	VN OP30000	FO	B0B1B4B6	III SUBJCT
SIGNAL		N01.00	SIGNAL-Y			00A-0814	NDI1M000	--------A--- ---------M---		803 OBJECT

A Segment of a Subordinate Clause

Fig. 5-16

left object, and predicate head prediction. Otherwise, no indication is made that the prediction has been fulfilled, and the testing is resumed with the next prediction. There is a second intersection between the newly activated subject prediction and the first alternative argument. Since, to the program, it seems that this is the first intersection, который is accepted as the subject of the clause. The reason for inserting the inactive predictions into the pool earlier is now evident. If the inactive predictions were not in the pool, it would be impossible to select который as the subject of the clause. The relative pronoun can be considered as fulfilling two independent functions, on the one hand, introducing a subordinate clause and, on the other hand, taking on an active role within the clause. In the procedure described, который activates the mechanism by which it itself is identified.

There are seven other intersections between the predictions in the pool and the alternative arguments of который, all of which are stored on the hindsight file. Since an adjectival alternative argument of который has been selected as the preferred argument, the prediction of a master is inserted at the top of the pool above the left object and predicate head predictions which have just been activated.

Since the next word позволяет is a verb, the master prediction is not fulfilled but is wiped from the pool and recorded on the hindsight file, and позволяет is accepted as the predicate of the clause. The wiped prediction is an indication that on the next pass through the sentence the nominal alternative argument of который should be selected as the preferred argument.

This technique is also effective when the relative pronoun is in an oblique case and is part of another independent nested structure, as in

..., в котором усреднение сигнала осуществляется... (Fig. 5-17). The preposition в is accepted by the infinity prediction, and the phraser, relative conjunction, and relative pronoun predictions are pushed down deeper into the pool when the new preposition complement predictions are entered at the top.

Both the adjectival and the nominal alternative arguments intersect with the locative singular preposition complement prediction, and the former alternative argument is selected as the preferred argument. The testing continues and when the relative pronoun prediction is fulfilled, the same activating process is carried out as when который was analyzed in the last example. Of course, this time котором cannot be accepted as the subject. The activated predictions are now in a position to make the selection of усреднение as the subject of the clause and осуществляется as the predicate of the clause.

A relative pronoun can occur in another manner that cannot be correctly analyzed on a single pass by the existing program. This format can be exemplified by the clause ..., конденсатор которого... (Fig. 5-18). When the alternative arguments of конденсатор are tested, there is no way of knowing that the relative pronoun которого will occur immediately after конденсатор, and that конденсатор is the subject of the clause. Very often, as in the example, the noun preceding the relative pronoun does not intersect with any of the existing predictions and its preferred argument is selected as arbitrary choice. The best that can be done in such a situation, in the framework of predictive syntactic analysis, is to recognize the relative pronoun when it finally occurs and preserve the necessary information for the analysis to be corrected on the next pass.

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO	SIZE OF POOL	TEXT SERIAL NO	ORGANIZED WORD	PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
*,		*,	00	19	00A-0446	,	0			INF COMMA
IN(TO)	I01.00	-V	00	30	00A-0447	R	--A--P--A--P		APOR00BA0650	INF PREP
WHICH	P01.00	KOTOR-OM	00	34	00A-0448	PA K STRITO	-----P------	-----B-------		447 R COMP
AVERAGING	N10.00	USREDNENI-E	00	33	00A-0449	NDI1N000	N-----------	N------------		446 SUBJCT
SIGNAL	N01.00	SIGNAL-A	00	26	00A-0450	NDI1M000	-G----------	-M-----------		449 N COMP
TO CARRY OUT	V01.00	OSUSHCHESTVL JA-ETSJA	00	29	00A-0451	VN OP70000	00T000BADR		B0B1B4B6	446 V PRED
MEANS	N08.00	SREDSTV-AMI	00	25	00A-0452	NDI1N000	----------I-	-----------N-		451 OBJECT
PULSE	A02.00	IMPUL,SN-OJ	00	25	00A-0453	AD00000	-G----------	-F-----------		452 N COMP
RADIO ENGINE ERING	N04.10	RADIOTEXNIK- I	00	28	00A-0454%	NDI1F100	-G----------	-F-----------		453 N COMPM

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO
*,		*,	00A-0446		*			
IN(TO)	I01.00	-V	00A-0447	R	--A--P--A--P		APOR00BA0650	000020000000
WHICH	P01.00	KOTOR-OM	00A-0448	PK K STRITO	-----P------	-----B-------		095097755098
AVERAGING	N10.00	USREDNENI-E	00A-0449	NDI1N000	N-A---------	N-N----------		206195000000
SIGNAL	N01.00	SIGNAL-A	00A-0450	NDI1M000	-G----------	-M-----------		183370000000
TO CARRY OUT	V01.00	OSUSHCHFSTVL JA-ETSJA	00A-0451	VN OP70000	--T---BADR		B0B1B4B6	130380000000
MEANS	N08.00	SREDSTV-AMI	00A-0452	NDI1N000	----------I-	-----------N-		191000000000
PULSE	A02.00	IMPUL,SN-OJ	00A-0453	AD00000	-G-CIP------	-F-FFF-------		078990000000
RADIO TECHNI CIAN	N01.10	RADIOTEXNIK- I	00A-04541	NDA1M000	-------N----	-------M-----		168720000000
RADIO ENGINE ERING	N04.10	RADIOTEXNIK- I	00A-0454%	NDI1F100	-G----N-A---	-F----F-F---		168730000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT		CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO	ORGANIZED WORD	ALTERNATIVE ARGUMENT		POTENTIAL ATTRIBUTED ARGUMENT
WHICH		P01.00	KOTOR-OM	00A-0448	PN K STRITO	-----P------	-----B-------	447 R COMP
PREDICTION	WIPED	----	448013000972					
AVERAGING		N10.00	USREDNENI-E	00A-0449	NDI1N000	--A---------	--N----------	446 L OBJ
MEANS		N08.00	SREDSTV-AMI	00A-0452	NDI1N000	----------I-	-----------N-	431 OBJECT
PULSE		A02.00	IMPUL,SN-OJ	00A-0453	AD00000	----I-------	----F--------	431 OBJECT
PREDICTION	WIPED	----	431011005927					

A Segment of a Subordinate Clause

Fig. 5-17

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT	3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
*,		*,	02	04	00A-0189	, O			INF COMMA
CAPACITOR	N01.00	KONDENSATOR-	03	15	00A-0190	NDI1M000	N---------- M----------		III ARBTR
WHICH	P01.00	KOTOR-OGO	03	10	00A-0191	PA K STRITO	-G--------- -B---------		190 N COMP
AND	I01.00	-I	03	12	00A-0192	CH			INF CONJNCT
TO CARRY OUT	V01.00	VYPOLNJA-ET	03	12	00A-0193	VN OP70000	00T000BADO	B0B1B4B6	189 V PRED
PROBLEM	N04.10	ZADACH-U	03	09	00A-0194	NDI4F000	--A-------- --F--------		193 OBJECT
STORAGE	N10.00	NAKOPLENI-JA	03	11	00A-0195	NDI1N000	-G--------- -N---------		194 N COMP

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
*,		*,	00A-0189				
CAPACITOR	N01.00	KONDENSATOR-	00A-0190	NDI1M000	N-A-------- M-M--------		091240000000
WHICH	P01.00	KOTOR-OGO	00A-0191	PK K STRITO	-GA-------- -BM--------		095073877549
AND	I01.00	-I	00A-0192	CH			000080000000
TO CARRY OUT	V01.00	VYPOLNJA-ET	00A-0193	VN OP70000	--T---BAD-	B0B1B4B6	035130000000
PROBLEM	N04.10	ZADACH-U	00A-0194	NDI4F000	--A-------- --F--------		060940000000
STORAGE	N10.00	NAKOPLENI-JA	00A-0195	NDI1N000	-G----N-A--- -N----N-N---		113260000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT	POTENTIAL ATTRIBUTED ARGUMENT
CAPACITOR	N01.00	KONDENSATOR-	00A-0190	NDI1M000	--A-------- --M--------	III ARBTR
WHICH	P01.00	KOTOR-OGO	00A-0191	PN K STRITO	-G--------- -B---------	190 N COMP
WHICH	P01.00	KOTOR-OGO	00A-0191	PA K STRITO	--A-------- --M--------	189 L OBJ
WHICH	P01.00	KOTOR-OGO	00A-0191	PN K STRITO	--A-------- --M--------	199 L OBJ
PREDICTION WIPED	----	191013000972				

A Segment of a Subordinate Clause

Fig. 5-18

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
*.		*.								
THAT	I01.00	CHT-O	00	08	00A-2352	. O				INF COMMA
TO ELIMINATE	V01.00	ISKLJUCHA-ET	00	19	00A-23531	O				352 R CONJ
FINING	N10.00	NALOZHENI-E	00	14	00A-2354	VN 0P30000	00T000BAD0		B0818486	352 V PRED
FLUCTUATION	N07.00	FLUKTUATSI-J	00	13	00A-2355	NDI1N000	--A---------	--N---------		354 OBJECT
			00	15	00A-2356	NDI1F000	-------B----	-------F----	T5	355 N COMP

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
*.		*.	00A-2352	*				
THAT	I01.00	CHT-O	00A-23531	O				
WHICH	P01.00	CHT-O	00A-2353%	PNCI STRI O	N-A---------	N-N---------		213848750000
TO ELIMINATE	V01.00	ISKLJUCHA-ET	00A-2354	VN OP300C0	--T---BAD-		B0818486	213847500000
FINING	N10.00	NALOZHENI-E	00A-2355	NDI1N000	N-A---------	N-N---------		081670000000
FLUCTUATION	N07.00	FLUKTUATSI-J	00A-2356	NDI1F000	-------B----	-------F----		113650000000
							T5	208780000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT		CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT			POTENTIAL ATTRIBUTED ARGUMENT
WHICH		P01.00	CHT-O	00A-2353%	PNCI STRI O	N-----------	N-----------		352 SUBJCT
WHICH		P01.00	CHT-O	00A-2353%	PNCI STRI O	--A---------	--N---------		352 L OBJ
FINING		N10.00	NALOZHENI-E	00A-2355	NDI1N000	N-----------	N-----------		352 SUBJCT
CONSTANT		A01.00	POSTOJANN-OM U	00A-2361	NDIOFB	---C--------	---B--------	T6	360 R COMP
READING		N01.00	OTSCH-ET	00A-2364	NDI1N000 1	N-----------	N-----------		352 SUBJCT
TEN		D01.00	DESJAT-I	00A-2365	DA RACJPK	-------B----	-------A----	0000000TOTOT	364 N COMP
TEN		D01.00	DESJAT-I	00A-2365	DN RACJPK	-G----------	-A----------	0000000TOTOT	364 N COMP
TEN		D01.00	DESJAT-I	00A-2365	DN RACJPK	-------B----	-------A----	0000000TOTOT	364 N COMP
AVERAGED		A01.00	USREDNENN-YX	00A-2366	AD0000 3	-----------P	-----------A		365 N COMPM
PREDICTION	WIPED	----	352012006900						

A Segment of a Subordinate Clause

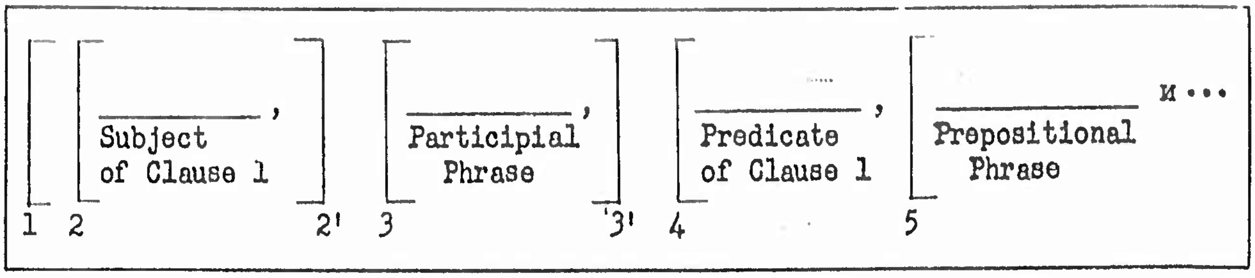
Fig. 5-19

There is one other problem in clause identification that will be discussed. Some words such as что can intersect with both the <u>relative conjunction</u> and <u>relative pronoun</u> predictions. If что is used as the relative pronoun, then it is also the subject or the object of the clause. Most of the time что is used as a relative conjunction, so that the prediction for <u>relative conjunction</u> is placed higher in the pool. If что is used as a relative pronoun and is the subject of the clause, no subject would be found in the clause as in the example ..., что исключает наложение ..., (Fig. 5-19). On the next pass, the relative pronoun alternative argument will be selected as the preferred argument.

8. The Conjunction и

Only one use of the conjunction и, namely its use as a link to connect two similar words, has been considered in the predictive syntactic analysis program so far. The linking property can be expressed completely in terms of predictive analysis. И may link the word following it with any word located in a nested structure that has not been completed. This is illustrated schematically in Fig. 5-20. Parentheses have been placed around every nested structure. The parentheses have been numbered and the right-parentheses have been marked with primes.

Preceding the и, the sentence has one clause indicated by left parenthesis 1. Within the clause, the subject noun phrase has been completely analyzed (parentheses 2 and 2'), while the predicate verb phrase is still open (parenthesis 4). A participial phrase has been completely analyzed (parentheses 3 and 3'), and a prepositional phrase that is part of



Schematic Representation of a Sentence with и

Fig. 5-20

the verb phrase is open (parenthesis 5). The word following и can be linked to either one of the words within the prepositional phrase or to one of the words in the verb phrase of the clause; however, the word following и cannot be linked to any word of the participial phrase or the subject noun phrase, since those nested structures have already been completed. If the link turns out to be with a word in the verb phrase, then the prepositional phrase is considered completely identified.

If a prediction for a possible link is made by inserting a prediction for a compound subject, compound noun complement, etc., into the prediction pool, then the addition of an end wipe prediction directly below the compound prediction identifies the nested structure. Since the compound prediction cannot be fulfilled except by a word following an и, the predictions are made inactive by means of PSI = 99. When an и is identified, these predictions are activated for a single cycle, that is, for the testing cycle of the next word. If an activated compound prediction is not fulfilled and is not wiped, then it is deactivated when the prediction pool is updated. This activation and deactivation process is carried out completely with the modification of the prediction span indicators. A PSI of 99 indicates that

the prediction is inactive and 49 indicates that the prediction has been activated for one testing cycle, after which it is reset to 99.

In the prediction pool, the compound predictions are ordered in the inverse order of occurrence. The last compound prediction made is the one nearest the top of the pool and will be tested for fulfillment first.

Several examples from text material will be presented indicating several types of problems involved in the identification of the linkage between the words.

A simple example of compounding is shown in the participial phrase: позволяющие выделить основную частоту периодического сигнала и измерить... (Fig. 5-21). The infinitive verb выделить is identified as the verb master of the participle позволяющие. The attributed argument makes a prediction of a compound verb master which is eventually activated when the conjunction и is identified. The word following the и is also a verb infinitive, измерить, and the only intersection is with the now activated compound verb master prediction. Meanwhile, the nested object noun phrase string, основную частоту периодического сигнала, has been recognized. Since the linkage of измерить goes beyond the object string to выделить, the identification of the noun phrase is completed.

Consider next the prepositional phrase: ...из цепочки запертых мультивибраторов и управляющей схемы (Fig. 5-22). The noun цепочки is identified as the preposition complement of из, after which the noun phrase, запертых мультивибраторов, is identified as the noun complement of цепочки. A compound preposition complement in the genitive case is predicted by the attributed argument of цепочки, and a compound noun complement (obviously in

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
DEVICE	N08.00	USTROJSTV-A	00	20	00A-0154	NDI1N000	-------N------	-------N------		III SUBJCT
*.		*.	00	11	00A-0155	. O				INF COMMA
ALLOWING	A04.00	POZVOLJAJUSH CH-IE	00	22	00A-0156	AD0100 4	-------N-A---	-------A-A---		155 FRASER
TO PICK OUT	V04.00	VYDELI-T.	00	25	00A-0157	VS OP30000	FO		BOB6	156 V MAST
FUNDAMENTAL	A02.00	OSNOVN-UJU	00	27	00A-0158	AD00000	--A----------	--F----------		157 OBJFCT
FREQUENCY	N04.00	CHASTOT-U	00	29	00A-0159	NDI2F000	--A----------	--F----------		158 OBJFCTM
PERIODIC(ALL Y)	A06.00	PERIODICHESK -OGO	00	29	00A-0160	AD0000	-G------------	-B------------		T7 159 N COMP
SIGNAL	N01.00	SIGNAL-A	00	32	00A-0161	NDI1M000	-G------------	-M------------		160 N COMPM
AND	I01.00	-I	00	32	00A-0162	CH				INF CONJNCT
TO MEASURE	V04.01	IZMERI-T.	00	31	00A-0163	VS OP30000	FO		BOB6	157 CV MAST

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO
DEVICE	N08.00	USTROJSTV-A	00A-0154	NDI1N000	-G----N-A---	-N----N-N---		206930000000
*.		*.	00A-0155					
ALLOWING	A04.00	POZVOLJAJUSH CH-IE	00A-0156	AD0100 4	-------N-A---	-------A-A---		149110000000
TO PICK OUT	V04.00	VYDELI-T.	00A-0157	VS OP30000	F-		BOB6	031912500000
FUNDAMENTAL	A02.00	OSNOVN-UJU	00A-0158	AD00000	--A----------	--F----------		129450000000
FREQUENCY	N04.00	CHASTOT-U	00A-0159	NDI2F000	--A----------	--F----------		212960000000
PERIODIC(ALL Y)	A06.00	PERIODICHESK -OGO	00A-0160	AD0000	-GA-----------	-BM-----------		T7 142210000000
SIGNAL	N01.00	SIGNAL-A	00A-0161	NDI1M000	-G------------	-M------------		183370000000
AND	I01.00	-I	00A-0162	CH				000080000000
TO MEASURE	V04.01	IZMERI-T.	00A-0163	VS OP30000	F-		BOB6	076580000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT		POTENTIAL ATTRIBUTED ARGUMENT
DEVICE	N08.00	USTROJSTV-A	00A-0154	NDI1N000	---------A---	---------N---	III L OBJ
ALLOWING	A04.00	POZVOLJAJUSH CH-IE	00A-0156	AD0100 4	---------A---	---------A---	III L OBJ
FUNDAMENTAL	A02.00	OSNOVN-UJU	00A-0158	AD00000	--A----------	--F----------	156 OBJECT
FUNDAMENTAL	A02.00	OSNOVN-UJU	00A-0158	AD00000	--A----------	--F----------	III L OBJ
FREQUENCY	N04.00	CHASTOT-U	00A-0159	NDI2F000	--A----------	--F----------	156 OBJECT
FREQUENCY	N04.00	CHASTOT-U	00A-0159	NDI2F000	--A----------	--F----------	III L OBJ
PERIODIC(ALL Y)	A06.00	PERIODICHESK -OGO	00A-0160	AD0000	--A----------	--M----------	T7 156 OBJECT
PERIODIC(ALL Y)	A06.00	PERIODICHESK -OGO	00A-0160	AD0000	--A----------	--M----------	T7 III L OBJ
AND	I01.00	-I	00A-0162	CH			000 R CONJ

A Segment of a Sentence

Fig. 5-21

```
                                           PREFERRED  ARGUMENT   AND  ATTRIBUTED  ARGUMENT

FIRST ENGLISH       CLASS    RUSSIAN WORD          CHAIN SIZE OF  TEXT        ORGANIZED                                                 3rd
 EQUIVALENT        MARKER  (TRANSLITERATED)        NO.   POOL   SERIAL NO.    WORD        PREFERRED  ARGUMENT                 SEMI-ORGANIZED  ATTRIBUTED
                                                                                                                                  WORD        ARGUMENT
IS                 V06.10  SOSTO-IT                 00   14   00A-28091   VN 3D600KO    00TOOOBADO                      C6 B1            III  V PRED
FROM               I01.00  IZ-                      00   08   00A-2810    R             -G------B----                      GOOROOA00300  INF PREP
CHAIN              N04.30  TSEPUCHK-I               00   09   00A-2811    NDI1F000      -G-----------  -F-----------                    810  R COMP
LOCKED             A03.00  ZAPERT-YX                00   11   00A-2812    AD0000    3   --------B----  --------A----                    811  N COMP
MULTIVIBRATO R     N01.00  MUL.TIVIBRAT OR-OV       00   14   00A-2813    NDI1M000      --------B----  --------M----                    812  N COMPM
AND                I01.00  -I                       00   14   00A-2814    CH                                                            INF CONJNCT
GOVERNING          A04.00  UPRAVLJAJUSH CH-EJ       00   13   00A-2815    AD0100    4   -G-----------  -F-----------                    812 CN COMP
CIRCUIT            N04.00  SXEM-Y                   00   16   00A-2816    NDI2FOYO      -G-----------  -F-----------                    815 CN COMPM
*.                         *.                       00   16   00A-2817    .           O                                                 END OF SENT.

                                                                      TEXTHADIC

                                                                                           CODING  DUE  TO                               DICTIONARY
                                                                                        WORD-BY-WORD ANALYSIS                            SERIAL  NO.
IS                 V06.10  SOSTO-IT                           00A-28091   VN 3D600KO    --T---BAD-                      C6 B1            189176666666
WILL TAKE PL ACE   V06.10  SOSTO-IT                           00A-2809%   VS 3000000    --T---CAD-                         B2            189180000000
FROM               I01.00  IZ-                                00A-2810    R             -G-----B----                       GOOROOA00300  074050000000
CHAIN              N04.30  TSEPOCHK-I                         00A-2811    NDI1F000      -G----N-A---   -F----F-F---                      212300000000
LOCKED             A03.00  ZAPERT-YX                          00A-2812    AD0000    3   --------BA--P  --------AA--A                     066075714284
MULTIVIBRATO R     N01.00  MUL.TIVIBRAT OR-OV                 00A-2813    NDI1M000      --------B----  --------M----                     110670000000
AND                I01.00  -I                                 00A-2814    CH                                                             000080000000
GOVERNING          A04.00  UPRAVLJAJUSH CH-EJ                 00A-2815    AD0100    4   -G-CIP------   -F-FFF------                      204530000000
CIRCUIT            N04.00  SXEM-Y                             00A-2816    NDI2FOYO      -G----N-A---   -F----F-F---                      194360000000
*.                         *.                                 00A-2817              *

                                                                      HINDSIGHT
                                                                                                                                         POTENTIAL
                                                                                                                                         ATTRIBUTED
                                                                                        ALTERNATIVE  ARGUMENT                            ARGUMENT
WILL TAKE PL ACE   V06.10  SOSTO-IT                           00A-2809%   VS 3000000    00TOOOCADO                         B2            III  V PRED
CHAIN              N04.30  TSEPOCHK-I                         00A-2811    NDI1F000      ---------A---  ---------F---                     809  OBJECT
LOCKED             A03.00  ZAPERT-YX                          00A-2812    AD0000    3   ---------A---  ---------A---                     809  OBJECT
GOVERNING          A04.00  UPRAVLJAJUSH CH-EJ                 00A-2815    AD0100    4   -G-----------  -F-----------                     811 CR COMP
CIRCUIT            N04.00  SXEM-Y                             00A-2816    NDI2FOYO      ---------A---  ---------F---                     809  OBJECT
PREDICTION   WIPED ----    809011001927
*.                         *.                                 00A-2817    .           O                                                 END OF SENT.
```

A Prepositional Phrase

Fig. 5-22

the genitive case) is predicted by the attributed argument of запертых. These two predictions are activated by the conjunction и.

When the four alternative arguments of управляющей are tested against the predictions in the pool, the two compound predictions are at the top of the pool. Since both predictions can be fulfilled by a genitive adjective alternative argument, the attributed argument is compound noun complement, while the second intersection of compound preposition complement is noted on the hindsight file. Finally, схемы is identified as the master of the compound noun complement управляющей. There is no way of determining on the basis of a syntactic analysis whether управляющей схемы is the compound noun complement or the compound preposition complement. A second pass would have to recognize that this ambiguity exists and list both interpretations as possible ones.

Another prepositional phrase brings up another interesting difficulty: ... приведут к нарушению компенсации равновероятных положительных и отрицательных выбросов шума... (Fig. 5-23). The noun нарушению is identified as the prepositional complement of к, after which the alternative arguments of компенсации are tested against the predictions in the pool. Because of the ordering of the predictions in the pool, the argument attributed to компенсации is the noun complement of нарушению, while the second possible intersection of object of the verb приведут is noted on the hindsight file. Here, once more, is an example of an ambiguous situation which cannot be resolved by means of a syntactic analysis alone.

In any event, the string равновероятных положительных is identified as part of a noun phrase acting as a noun complement of компенсации. At this time the following predictions are at the top of the pool:

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
TO HAPPEN	V08.20	PRIVED-UT	00	39	00A-12651	VS KP20000	00000TCADO		B2B4	III V PRED
TO	I01.00	K-	00	08	00A-1266	R	---C------C--		C00R00A00300	INF PREP
FAULT	N10.00	NARUSHENI-JU	00	09	00A-1267	NDI1N000	---C---------	---N---------		266 R COMP
COMPENSATION	N07.00	KOMPENSATSI- I	00	11	00A-1268	NDI1F000	-G-----------	-F-----------		267 N COMP
EQUIPROBAPLE	A02.00	RAVNOVEROJAT N-YX	00	14	00A-1269	AD01000	--------G----	--------A----		268 N COMP
POSITIVE	A02.00	POLOZHITEL,N -YX	00	17	00A-1270	AD00000	--------G----	--------A----		269 N COMPM
AND	I01.00	-I	00	17	00A-1271	CH				INF CONJNCT
NEGATIVE	A02.00	OTRITSATEL,N -YX	00	17	00A-1272	AD00000	--------G----	--------A----		270 N COMPM
BLOWOUT	N01.00	VYBPOS-OV	00	17	00A-1273	NDI1M000	--------G----	--------M----		272 N COMPM
NOISE	N01.00	SHUM-A	00	17	00A-1274	NDI1M000	-G-----------	-M-----------		273 N COMP

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS			SEMI-ORGANIZED WORD		DICTIONARY SERIAL NO.
TO HAPPEN	V08.20	PRIVED-UT	00A-12651	VS KP20000	-----TCAD-			B2B4		157625000000
WILL BRING	V08.20	PRIVED-UT	00A-1265%	VS00P300K0	-----TCAD-		P309	B2B4B5	T4	157790000000
TO	I01.00	K-	00A-1266	R	---C------C--			C00R00A00300		084690000000
FAULT	N10.00	NARUSHENI-JU	00A-1267	NDI1N000	---C---------	---N---------				114980000000
COMPENSATION	N07.00	KOMPENSATSI- I	00A-1268	NDI1F000	-G-C-PN-A----	-F-F-FF-F---				090610000000
EQUIPROBAPLE	A02.00	RAVNOVEROJAT N-YX	00A-1269	AD01000	--------GA--P	--------AA--A				166817500000
POSITIVE	A02.00	POLOZHITEL,N -YX	00A-1270	AD00000	--------GA--P	--------AA--A				150470000000
AND	I01.00	-I	00A-1271	CH						000080000000
NEGATIVE	A02.00	OTRITSATEL,N -YX	00A-1272	AD00000	--------GA--P	--------AA--A				133870000000
BLOWOUT	N01.00	VYBROS-OV	00A-1273	NDI1M000	--------G----	--------M----				030550000000
NOISE	N01.00	SHUM-A	00A-1274	NDI1M000	-G-----------	-M-----------				215870000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT					POTENTIAL ATTRIBUTED ARGUMENT
WILL BRING	V08.20	PRIVED-UT	00A-1265%	VS00P300K0	00000TCADO		P309	B2B4B5	T4	III V PRED
FAULT	N10.00	NARUSHENI-JU	00A-1267	NDI1N000	---C---------	---N---------				265 OBJECT
COMPENSATION	N07.00	KOMPENSATSI- I	00A-1268	NDI1F000	---C---------	---F---------				265 OBJECT
NEGATIVE	A02.00	OTRITSATEL,N -YX	00A-1272	AD00000	--------G----	--------A----				269 CN COMP
NEGATIVE	A02.00	OTRITSATEL,N -YX	00A-1272	AD00000	--------G----	--------A----				268 CN COMP

A Segment of a Sentence

Fig. 5-23

(1) Master (of положительных)

(2) Compound noun complement (of равновероятных)

(3) Compound noun complement (of компенсации)

(4) Compound preposition complement (of нарушению)

(5) Object (of приведут)

The first of the three alternative arguments of отрицательных intersect with the first three predictions in the pool, and the attributed argument is master of положительных. Since the noun phrase has not been completed, отрицательных is linked either with равновероятных or положительных. However, the distinction cannot be drawn on syntactic lines and a mistake can occur.

The remainder of the examples will be concerned with compound subjects and predicates. The first example is a compound predicate following the subject: ... импульсы формируются буферными лампами и подаются ... (Fig. 5-24). Подаются agrees with формируются in person, number, tense, and voice, and thereby intersects with the compound predicate head prediction generated by the attributed argument of формируются. The object noun phrase, буферными лампами, is then identified, and the process is terminated.

A somewhat more interesting example is: ... что синхронный фильтр пригоден ... и дает ... (Fig. 5-25), in which the indicative verb дает is compounded with the short form adjective пригоден, which is acting as the predicate of the clause.

If the subject is a compound one, and if the predicate head prediction has already been modified to accept only a singular predicate, then it is necessary to modify the predicate head prediction again, so that

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT	3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
NEGATIVE	A02.00	OTRITSATEL,N -YE	00	20	00A-3179	AD00000	-------N------ -------A------		III SUBJCT
GOVERNING	A04.00	UPRAVLJAJUSH CH-IE	00	11	00A-3180	AD0100 4	-------N------ -------A------		179 SUBJCTM
PULSE	N01.00	IMPUL,S-Y	00	11	00A-3181	NDI1M000	-------N------ -------M------		180 SUBJCTM
SHAPE	V03.00	FORMIRU-JUTS JA	00	11	00A-3182	VN 0P70000	00000TBADR	B1B4B5	III V PRED
BUFFER	A02.00	BUFERN-YMI	00	08	00A-3183	AD00000	-----------I- -----------A-		182 OBJECT
TUBE	N04.00	LAMP-AMI	00	10	00A-3184	NDI2F000	-----------I- -----------F-		183 OBJECTM
AND	I01.00	-I	00	10	00A-3185	CH			INF CONJNCT
MOVE	V12.00	PODA-JUTSJA	00	09	00A-3186	VNR0000000	00000TBADR	B1	182 CV PRED

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
NEGATIVE	A02.00	OTRITSATEL,N -YE	00A-3179	AD00000	-------N-A--- -------A-A---		133870000000
GOVERNING	A04.00	UPRAVLJAJUSH CH-IE	00A-3180	AD0100 4	-------N-A--- -------A-A---		204530000000
PULSE	N01.00	IMPUL,S-Y	00A-3181	NDI1M000	-------N-A--- -------M-M---		078940000000
SHAPE	V03.00	FORMIRU-JUTS JA	00A-3182	VN 0P70000	-----TBADR	B1B4B5	209450000000
BUFFER	A02.00	BUFERN-YMI	00A-3183	AD00000	-----------I- -----------A-		010670000000
TUBE	N04.00	LAMP-AMI	00A-3184	NDI2F000	-----------I- -----------F-		098680000000
AND	I01.00	-I	00A-3185	CH			000080000000
MOVE	V12.00	PODA-JUTSJA	00A-3186	VNR0000000	-----TBADR	B1	145790000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT	POTENTIAL ATTRIBUTED ARGUMENT
NEGATIVE	A02.00	OTRITSATEL,N -YE	00A-3179	AD00000	---------A--- ---------A---	III L OBJ
GOVERNING	A04.00	UPRAVLJAJUSH CH-IE	00A-3180	AD0100 4	---------A--- ---------A---	III L OBJ
PULSE	N01.00	IMPUL,S-Y	00A-3181	NDI1M000	---------A--- ---------M---	III L OBJ

A Segment of a Sentence

Fig. 5-24

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
*.		*.	02	05	00A-2755	. 0				INF COMMA
THAT	I01.00	CHT-O	02	16	00A-27561	C				755 R CONJ
SYNCHRONOUS	A01.00	SINXRONN-YJ	02	11	00A-2757	AD00000	N-----------	M-----------		755 SUBJCT
FILTER	N01.00	FIL.TR-	02	12	00A-2758	NDI1MOYO	N-----------	M-----------		757 SUBJCTM
FIT	A02.00	PRIGODEN-	02	12	00A-2759	AD00000	N-----------	M-----------		755 A PRED
FOR	I01.00	DL-JA	02	07	00A-2760	R	-G-----G----		G00R00600400	INF PREP
QUANTITATIVE	A01.00	KOLICHESTVEN N-YX	02	09	00A-2761	AD01000	-------G----	-------A----		760 R COMP
MEASUREMENT	N10.00	IZMERENI-J	02	11	00A-2762	NDI1N000	-------G----	-------N----		761 R COMPM
IN(TO)	I01.00	-V	02	11	00A-2763	R	--A--P--A--P		APOR00BAU650	INF PREP
DEFINITE	A01.00	OPREDELENN-O M	02	14	00A-2764	AD0000 3	-----P------	-----B------		763 R COMP
RANGE	N01.00	DIAPAZON-E	02	14	00A-2765	NDI1M000	-----P------	-----M------		764 R COMPM
FREQUENCY	N04.00	CHASTOT-	02	14	00A-2766	NDI2F000	-------G----	-------F----		765 N COMP
AND	I01.00	-I	02	17	00A-2767	CH				INF CONJNCT
GIVE	V12.00	DA-ET	02	16	00A-2768	VN 0000000	00T000BADO		B1	759 CV PRED
IMPROVEMENT	N10.00	ULUCHSHENI-F	02	09	00A-2769	NDI1N100	--A---------	--N---------		768 OBJECT

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
*.		*.	00A-2755		*			
THAT	I01.00	CHT-O	00A-27561	C				213848750000
WHICH	P01.00	CHT-O	00A-2756%	PNCI STRI 0	N-A---------	N-N---------		213847500000
SYNCHRONOUS	A01.00	SINXRONN-YJ	00A-2757	AD00000	N-A---------	M-M---------		184400000000
FILTER	N01.00	FIL.TR-	00A-2758	NDI1MOYO	N-A---------	M-M---------		208510000000
FIT	A02.00	PRIGODEN-	00A-2759	AD00000 1	N-----------	M-----------		158320000000
FOR	I01.00	DL-JA	00A-2760	R	-G-----G----		G00P00600400	051970000000
QUANTITATIVE	A01.00	KOLICHESTVEN N-YX	00A-2761	AD01000	-------GA--P	-------AA--A		089830000000
MEASUREMENT	N10.00	IZMERENI-J	00A-2762	NDI1N000	-------G----	-------N----		076520000000
IN(TO)	I01.00	-V	00A-2763	R	--A--P--A--P		APOR00BAU650	000020000000
DEFINITE	A01.00	OPREDELENN-O M	00A-2764	AD0000 3	-----P------	-----B------		127730000000
RANGE	N01.00	DIAPAZON-E	00A-2765	NDI1M000	-----P------	-----M------		050330000000
FREQUENCY	N04.00	CHASTOT-	00A-2766	NDI2F000	-------G----	-------F----		212960000000
AND	I01.00	-I	00A-2767	CH				000080000000
GIVE	V12.00	DA-ET	00A-2768	VN 0000000	--T---BAD-		61	045030000000
IMPROVEMENT	N10.00	ULUCHSHENI-F	00A-2769	NDI1N100	N-A---------	N-N---------		203660000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT		POTENTIAL ATTRIBUTED ARGUMENT
WHICH	P01.00	CHT-O	00A-2756%	PNCI STRI 0	--A---------	--N---------	755 L OBJ
WHICH	P01.00	CHT-O	00A-2756%	PNCI STRI 0	--A---------	--N---------	754 OBJECT
SYNCHRONOUS	A01.00	SINXRONN-YJ	00A-2757	AD00000	--A---------	--M---------	755 L OBJ
SYNCHRONOUS	A01.00	SINXRONN-YJ	00A-2757	AD00000	--A---------	--M---------	754 OBJECT
FILTER	N01.00	FIL.TR-	00A-2758	NDI1MOYO	--A---------	--M---------	755 L OBJ
FILTER	N01.00	FIL.TR-	00A-2758	NDI1MOYO	--A---------	--M---------	754 OBJECT
QUANTITATIVE	A01.00	KOLICHESTVEN N-YX	00A-2761	AD01000	--------A---	--------A---	754 OBJECT
IMPROVEMENT	N10.00	ULUCHSHENI-F	00A-2769	NDI1N100	--A---------	--N---------	754 OBJECT

A Segment of a Subordinate Clause

Fig. 5-25

the prediction can now accept a plural predicate only. For example, in: отсутствие нелинейных эффектов и постоянство...обеспечиваются... (Fig. 5-26), when отсутствие is recognized as the subject of the clause, the predicate head prediction is modified, so that a predicate must agree with the subject in number. When постоянство is later recognized as the compound subject, the prediction is modified once more, so that it can only accept a plural predicate such as обеспечиваются.

Although и and или have been temporarily grouped together, it is obvious that in the above case they must be treated differently. If the clause had read ...или постоянство instead of ...и постоянство, then the predicate head should not have been modified to accept a plural predicate.

If the predicate precedes the compound subject, it is usual to find the predicate agreeing with the first subject as in the example, ...является анализ...и рассмотрение (Fig. 5-27). It is possible to find a preceding predicate written in the plural rather than the singular. This possibility will have to be incorporated in the predictive syntactic analysis program.

9. Summary

Several of the broad problem areas of syntactic analysis which have been included in the experimental program have been discussed. These problem areas basically are concerned with the syntactic relationships among individual words within phrases and clauses, as opposed to the syntactic relationships among the phrases and clauses themselves. In this section, a number

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT		3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
ABSENCE	N10.00	OTSUTSTVI-E	00	20	00A-1706	NDI1N000	N-----------	N-----------		III SUBJCT
NONLINEAR	A02.00	NELINEJN-YX	00	11	00A-1707	AD00000	-------G----	--------A---		706 N COMP
EFFECT	N01.00	EHFFEKT-OV	00	14	00A-1708	NDI1M000	-------G----	--------M---		707 N COMPH
AND	I01.00	-I	00	14	00A-1709	CH				INF CONJNCT
CONSTANCY	N08.00	POSTOJANSTV- O	00	13	00A-1710	NDI1N100	N-----------	N-----------		706 SUBJCT
FACTOR	N01.00	KOEHFFITSIEN T-A	00	13	00A-1711	NDI1M000	-G----------	-M----------		710 N COMP
AMPLIFICATIO N	N10.00	USILENI-JA	00	16	00A-1712	NDI1N1Y0	-G----------	-N----------		711 N COMP
AVERAGING	A04.00	USREDNJAJUSH CH-EGO	00	19	00A-1713	ADC0000	-G----------	-B----------		712 N COMP
DEVICE	N08.00	USTROJSTV-A	00	22	00A-1714	NDI1N000	-G----------	-N----------		713 N COMPH
TO PROVIDE	V01.00	OBESPECHIVA- JUTSJA	00	22	00A-1715	VN OP30000	00000TBADR		B0B19486	III V PRED

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
ABSENCE	N10.00	OTSUTSTVI-E	00A-1706	NDI1N000	N-A---------	N-N---------		13448000000
NONLINEAR	A02.00	NELINEJN-YX	00A-1707	ADC0000	-------GA--P	--------AA--A		11728000000
EFFECT	N01.00	EHFFEKT-OV	00A-1708	NDI1M000	-------G----	--------M---		21905000000
AND	I01.00	-I	00A-1709	CH				00008000000
CONSTANCY	N08.00	POSTOJANSTV- O	00A-1710	NDI1N100	N-A---------	N-N---------		15301000000
FACTOR	N01.00	KOEHFFITSIEN T-A	00A-1711	NDI1M000	-G----------	-M----------		09551000000
AMPLIFICATIO N	N10.00	USILENI-JA	00A-1712	NDI1N1Y0	-G----N-A---	-N----N-N---		20524000000
AVERAGING	A04.00	USREDNJAJUSH CH-EGO	00A-1713	AD00000	-GA---------	-BM---------		20624333333
DEVICE	N08.00	USTROJSTV-A	00A-1714	NDI1N000	-G----N-A---	-N----N-N---		20693000000
TO PROVIDE	V01.00	OBESPECHIVA- JUTSJA	00A-1715	VN OP30000	-----TBADR		B0B18486	12089000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT		POTENTIAL ATTRIBUTED ARGUMENT
ABSENCE	N10.00	OTSUTSTVI-E	00A-1706	NDI1N000	--A---------	--N---------	III L OBJ
NONLINEAR	A02.00	NELINEJN-YX	00A-1707	AD00000	---------A---	---------A---	III L OBJ
CONSTANCY	N08.00	POSTOJANSTV- O	00A-1710	NDI1N100	--A---------	--N---------	III L OBJ
AMPLIFICATIO N	N10.00	USILENI-JA	00A-1712	NDI1N1Y0	---------A---	---------N---	III L OBJ
AVERAGING	A04.00	USREDNJAJUSH CH-EGO	00A-1713	AD00000	--A---------	--M---------	III L OBJ
DEVICE	N08.00	USTROJSTV-A	00A-1714	NDI1N000	---------A---	---------N---	III L OBJ

A Segment of a Sentence

Fig. 5-26

PREFERRED ARGUMENT AND ATTRIBUTED ARGUMENT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	CHAIN NO.	SIZE OF POOL	TEXT SERIAL NO.	ORGANIZED WORD	PREFERRED ARGUMENT	3rd SEMI-ORGANIZED WORD	ATTRIBUTED ARGUMENT
OPJECT	N01.00	PREDMET-OM	00	20	00A-0428	NDI1M000	----I-------- ----M--------		III L OBJ
PRESENT	A04.00	NASTOJASHCH- EGO	00	11	00A-0429	AD00000	-G----------- -B-----------		428 N COMP
COMMUNICATIO N	N10.00	SOOBSHCHENI- JA	00	14	00A-0430	NDI1M000	-G----------- -N-----------		429 N COMPM
TO BE	V01.00	JAVLJA-FTSJA	00	14	00A-0431	VN 0000000	00T000BADR	80818486	III V PRED
ANALYSIS	N01.00	ANALIZ-	00	09	00A-0432	NDI1M000	N------------ M------------		III SUBJCT
POSSIBILITY	N06.00	VOZMOZHNOST- EJ	00	11	00A-0433	NDI1F000	--------G---- --------F----		432 N COMP
IMPROVEMENT	N10.00	ULUCHSHENI-J A	00	14	00A-0434	NDI1N100	-G----------- -N-----------		433 N COMP
RATIO	N10.00	OTNOSHENI-JA	00	17	00A-0435	NDI1M000	-G----------- -N-----------		434 N COMP
SIGNAL-TO-NO ISE	I01.00	SIGNAL-SHUM-	00	20	00A-0436	NU	-G----------- - ----------		435 N COMP
BY MEANS OF	I01.00	PUT-EM	00	23	00A-04371	H			INF ADVB
AVERAGING	N10.00	USREDNENI-JA	00	23	00A-0438	NDI1N000	-G----------- -N-----------		436 N COMP
PERIODIC(ALL Y)	A06.00	PERIODICHESK -OGO	00	26	00A-0439	AD0000	-G----------- -B-----------	T7	438 N COMP
SIGNAL	N01.00	SIGNAL-A	00	29	00A-0440	NDI1M000	-G----------- -M-----------		439 N COMPM
AND	I01.00	-I	00	29	00A-0441	CH			INF CONJNCT
EXAMINATION	N10.00	RASSMOTRENI- E	00	28	00A-0442	NDI1N100	N------------ N------------		432 CSUBJCT
CONCRETE	A02.00	KONKRETN-OJ	00	17	00A-0443	AD00000	-G----------- -F-----------		442 N COMP
CIRCUIT	N04.00	SXEM-Y	00	16	00A-0444	NDI2FOYO	-G----------- -F-----------		443 N COMPM
DEVICE	N01.00	PRIBOR-A	00	16	00A-0445	NDI1M000	-G----------- -M-----------		444 N COMP

TEXTHADIC

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
UPJECT	N01.00	PREDMET-OM	00A-0428	NDI1M000	----I-------- ----M--------		155030000000
PRESENT	A04.00	NASTOJASHCH- EGO	00A-0429	AD00000	-GA---------- -BM----------		115260000000
COMMUNICATIO N	N10.00	SOOBSHCHENI- JA	00A-0430	NDI1N000	-G----N-A---- -N-----N-N---		187950000000
TO BE	V01.00	JAVLJA-FTSJA	00A-0431	VN 0000000	--T---BADR	80818486	219270000000
ANALYSIS	N01.00	ANALIZ-	00A-0432	NDI1M000	N-A---------- M-M----------		002500000000
POSSIBILITY	N06.00	VOZMOZHNOST- EJ	00A-0433	NDI1F000	--------G---- --------F----		021500000000
IMPROVEMENT	N10.00	ULUCHSHENI-J A	00A-0434	NDI1N100	-G----N-A---- -N-----N-N---		203660000000
RATIO	N10.00	OTNOSHENI-JA	00A-0435	NDI1N000	-G----N-A---- -N-----N-N---		132780000000
SIGNAL-TO-NO ISE	I01.00	SIGNAL-SHUM-	00A-0436	NU	NGACIPNGACIP AAAAAA		183695000000
BY MEANS OF	I01.00	PUT-EM	00A-04371	H			166210000000
BY MEANS OF	I01.00	PUT-EM	00A-0437%	RH	-G------G----	G00HR0100100	166215000000
AVERAGING	N10.00	USREDNENI-JA	00A-0438	NDI1N000	-G----N-A---- -N-----N-N---		206195000000
PERIODIC(ALL Y)	A06.00	PERIODICHESK -OGO	00A-0439	AD0000	-GA---------- -BM----------	T7	142210000000
SIGNAL	N01.00	SIGNAL-A	00A-0440	NDI1M000	-G----------- -M-----------		183370000000
AND	I01.00	-I	00A-0441	CH			000080000000
EXAMINATION	N10.00	RASSMOTRENI- E	00A-0442	NDI1N100	N-A---------- M-M----------		174140000000
CONCRETE	A02.00	KONKRETN-OJ	00A-0443	AD00000	-G-CIP------- -F-FFF-------		091510000000
CIRCUIT	N04.00	SXEM-Y	00A-0444	NDI2FOYO	-G----N-A---- -F-----F-F---		194360000000
DEVICE	N01.00	PRIBOR-A	00A-0445	NDI1M000	-G----------- -M-----------		157740000000

HINDSIGHT

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	ALTERNATIVE ARGUMENT	3rd SEMI-ORGANIZED WORD	POTENTIAL ATTRIBUTED ARGUMENT
COMMUNICATIO N	N10.00	SOOBSHCHENI- JA	00A-0430	NDI1N000	-------N----- -------N-----		III SUBJCT
SIGNAL-TO-NO ISE	I01.00	SIGNAL-SHUM-	00A-0436	NU	--------G---- --------A----		435 N COMP
SIGNAL-TO-NO ISE	I01.00	SIGNAL-SHUM-	00A-0436	NU	----I-------- ---- --------		431 OBJECT
SIGNAL-TO-NO ISE	I01.00	SIGNAL-SHUM-	00A-0436	NU	-----------I- -----------A-		431 OBJECT
BY MEANS OF	I01.00	PUT-EM	00A-0437%	RH	-G------G----	G00HR0100100	INF PREP
CONCRETE	A02.00	KONKRETN-OJ	00A-0443	AD00000	----I-------- ----F--------		431 OBJECT

A Segment of a Sentence

Fig. 5-27

of other areas that either are being studied at this time or might be studied in the near future are mentioned.

In the present experimental program, the government of oblique cases by nouns, adjectives, and verbs has been largely neglected. To some extent this has been due to a previous deficiency of dictionary coding, which has only recently been rectified.[8]

Another important area that is being considered is the negative, in particular, the word не. If не occurs immediately preceding a transitive verb, the object governed by the verb can be in the genitive case rather than the accusative case. However, if an adverb intervenes between the verb and the object, the object must remain in the accusative case.

The first attempts at predicting entire phrases and clauses are being made. After a comparative adverb, a clause or phrase starting with чем will be predicted. Every noun will predict a modifier, a prediction that is normally inactive but which is activated after a comma. This prediction will be fulfilled by a participle which occurs after the comma and agrees with the noun in case and number. This same mechanism might prove useful in the identification of series of words separated by commas. In the near future, some prepositional phrases should be predicted by verbs. It is estimated that approximately half of the prepositional phrases that are found could be tied in this manner to a verb that they modify.

The broadest and most important area for future research is in the organization of correcting passes. These are necessary, on the one hand, to correct errors discovered during the first pass and, on the other hand, to establish the syntactic relationships among the phrases and clauses.

For the reader who wishes to try to analyze Russian sentences syntactically by means of the technique of predictive analysis, several sentences have been provided in Appendix F. All the words in these sentences have been analyzed on a word-by-word basis (see Chap. 3). The grammatical codes necessary to carry out the syntactic analysis are listed in Tables 3-4, 3-5, 3-7, 3-8, 3-9, and 3-12. The complete details of the coding in the texthadic items can be found in Ref. 8.

REFERENCES[†]

1. Bossert, W. H., "The Implementation of Predictive Analysis," NSF-4, Sec. VIII (1960).

2. Giuliano, V. E., "An Experimental Study of Automatic Language Translation," Doctoral Thesis, Harvard University (1959).

3. Yngve, V. H., "The Depth Hypothesis," Symposium on the Structure of Language and its Mathematical Aspects, 567th Meeting of the American Mathematical Society, New York (April 1960) (to appear in Proceedings of the Symposium, American Mathematical Society, Providence, Rhode Island).

4. Coppinger, L., "Bibliography of Recorded Russian Texts," NSF-4, Sec. II (1960).

5. Chomsky, N., Syntactic Structures, Mouton and Co., The Hague (1957).

6. Frink, O. and Kline, G., "Statistics on Grammatical Interpretations," NSF-3, Sec. III (1959).

7. Oettinger, A. G., "Automatic Syntactic Analysis and the Pushdown Store," Symposium on the Structure of Language and its Mathematical Aspects, 567th Meeting of the American Mathematical Society, New York (April 1960) (to appear in Proceedings of the Symposium, American Mathematical Society, Providence, Rhode Island).

8. Coppinger, L. and von Susich, S., "Grammatical Coding," NSF-4, Sec. III (1960).

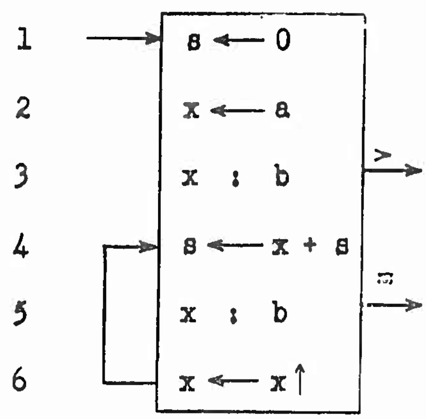
[†] In this bibliography the following abbreviation is used:
NSF-3, 4, etc. - Mathematical Linguistics and Automatic Translation, Report to the National Science Foundation. The Computation Laboratory of Harvard University, Cambridge, Massachusetts.

Appendix A

NOTATION FOR SEQUENTIAL OPERATIONS

Iverson[1,2] has proposed a new notation for sequential operations that is extremely useful over a wide range of data processing problems. The difficulty in expressing logical processes of automatic translation in classical forms of representation led to the adoption of this more powerful notation with minor modifications. Iverson's notation, as used in this report, is presented in this appendix. Only the operations used in this report are given.

A representative set of sequential operations is illustrated in Fig. A-1



Sum of Integers from a to b
Fig. A-1

Each line of the set of operations is a <u>step</u>, a specification of some quantity or quantities in terms of some finite operation upon a specified set of operands. Thus, "s is specified by the sum of the contents of s and x" is denoted by step 4. At certain branch points in the program more than one alternate step is specified as a possible successor. One of these

possible successors is chosen according to criteria determined in the step preceding the branch. The branch is denoted by a set of arrows leading to each possible successor step, and each arrow is labeled by the condition under which the corresponding successor is chosen. The step following the branch step is selected if none of the labeled conditions is met. In addition, any unlabeled arrow is always considered an unconditional transfer. Thus, in step 5, if x is equal to "b", the arrow is followed, and the process terminates. However, if x is not equal to "b", the process continues to step 6. After performing step 6, the operation always returns to step 4.

Consider the program in Fig. A-1. It is a representation for the program to sum all the integers from "a" to "b". In steps 1 and 2, s and x are initialized to "0" and "a" respectively. If "a" > "b", the program terminates at step 3. The adding operation takes place in step 4. If x is not equal to "b" in step 5, the program continues to step 6, after which it returns to step 4. The symbol, $x\uparrow$, represents the first successor of x, that is, x + 1. (A similar symbol, $x\downarrow$, represents the first predecessor of x.) This process is continued until x = "b", when the arrow terminating the program will be followed.

Since zero occurs frequently in comparisons, it is convenient to omit it. Thus, if a variable stands alone at a branch point, comparison with zero is implied. Moreover, since comparisons on an index frequently occur immediately after it is modified, a branch at a point of modification will denote branching on the indicated index, the comparison occurring <u>after</u> modification.

Scalars, vectors, and matrices will be used in the operations and will be indicated, respectively, by lower case letters (x), lower case letters

underlined ($\underline{x}$), and upper case letters underlined ($\underline{X}$). Components of a row vector will be indicated by a subscript, x_i, while components of a column vector will be indicated by a superscript, x^i. A vector will be assumed to be a row vector unless otherwise specified. Rows of a matrix will be designated by a superscript, X^i, and columns by a subscript, X_j. One general restriction on operations is the need for compatibility of the operands. Compatibility conditions (shown in column 4 of Table A-1) concern the dimensions of the operands ($\nu(\underline{x})$, $\nu(\underline{X})$, $\mu(\underline{X})$) and in most cases the dimensions must be equal.

The <u>weight</u> of a <u>logical vector</u> $\underline{x}$, that is, a vector every component of which is a "1" or a "0", is denoted by $\sigma(\underline{x})$ and is defined as the number of unit components in $\underline{x}$. The logical <u>head</u> vector h^j contains a "1" in the first j positions, and the logical <u>tail</u> vector t^j contains a "1" in the last j positions. A logical vector with but one "1" in the j^{th} position is denoted by $\underline{\epsilon}^j$, and a logical vector $\underline{x}$ with $\sigma(\underline{x}) = \nu(\underline{x})$ is denoted by $\underline{\epsilon}$.

A list of the operations that are used in this report follows. The operations are summarized in Table A-1.

1. <u>Scalar replacement</u>: If c is a scalar and $\underline{u}$ is a logical vector and $\underline{x} = c\underline{u}$, then $x_i = cu_i$. Thus, if $\underline{u} = [1,0,1,0,1]$, then $\underline{x} = [c,0,c,0,c]$.

2. <u>Negation</u>: If $\bar{\underline{u}}$ is the negation of $\underline{u}$, then $\bar{u}_i = 0$ if $u_i = 1$ and $\bar{u}_i = 1$ if $u_i = 0$. In the example of 1, $\underline{a} = c\bar{\underline{u}} = [0,c,0,c,0]$.

3. <u>Logical sum</u>: If $\underline{w} = \underline{u} \vee \underline{v}$, then $w_i = u_i \vee v_i$. $\nu(\underline{w}) = \nu(\underline{u}) = \nu(\underline{v})$. Thus, if $\underline{u} = [1,0,1,0,1]$ and $\underline{v} = [0,0,1,1,0]$, then $\underline{w} = [1,0,1,1,1]$.

4. Logical product: If $\underline{w} = \underline{u} \wedge \underline{v}$, then $w_i = u_i \wedge v_i$ with the same compatibility conditions as in 3. Using the same $\underline{u}$ and $\underline{v}$, $\underline{w} = [0,0,1,0,0]$.

5. Compression: If $\underline{x} = \underline{u}/\underline{y}$, then the vector $\underline{x}$ contains only those components of the vector $\underline{y}$ for which $u_i = 1$, and $\underline{x}$ is ordered on $\underline{y}$. $\nu(\underline{y}) = \nu(\underline{u})$. If $\underline{u} = [1,0,1,0,1]$ and $\underline{y} = [y_1,y_2,y_3,y_4,y_5]$, then $\underline{x} = [y_1,y_3,y_5]$.

The compression operation is extended to matrices as follows: a row compression, denoted by $\underline{u}/\underline{X}$, compresses each row vector $\underline{X}^i$ of the matrix $\underline{X}$ to form a new matrix of dimension $\mu(\underline{X}) \times \sigma(\underline{u})$. Column compression, denoted by $\underline{u} /\!\!/ \underline{X}$, compresses each column vector $\underline{X}_i$ to form a matrix of dimension $\sigma(\underline{u}) \times \nu(\underline{X})$. Compatibility conditions are $\nu(\underline{u}) = \nu(\underline{X})$ for row compression and $\nu(\underline{u}) = \mu(\underline{X})$ for column compression.

In the event of compression by a logical head or tail vector of a vector $\underline{x}$ such that $\sigma(\underline{h})$ or $\sigma(\underline{t}) > \nu(\underline{x})$, the resultant is defined such that $\underline{h}/\underline{x} = \underline{x}$ and $\underline{t}/\underline{x} = \underline{x}$.

6. Mesh: Given the vectors $\underline{x}$ and $\underline{y}$ and the logical vector $\underline{u}$, the mesh of $\underline{x}$ and $\underline{y}$ under the control of $\underline{u}$, $\backslash \underline{x}, \underline{u}, \underline{y} \backslash$, results in a vector $\underline{z}$ such that $\underline{u}/\underline{z} = \underline{y}$ and $\overline{\underline{u}}/\underline{z} = \underline{x}$. $\nu(\underline{u}) = \nu(\underline{z})$, $\sigma(\underline{u}) = \nu(\underline{y})$ and $\sigma(\overline{\underline{u}}) = \nu(\underline{x})$. If $\underline{u} = [1,0,1,0,1]$, $\underline{x} = [2,3]$, and $\underline{y} = [7,8,9]$, then $\underline{z} = \backslash \underline{x}, \underline{u}, \underline{y} \backslash = [7,2,8,3,9]$.

7. Logical reduction: If $\underline{v} = \underline{x}\ R\ \underline{y}$, then $v_i = 1 \longleftrightarrow x_i\ R\ y_i$. Any relation can be substituted for "R". In particular, if "R" is "=", then for $\underline{x} = [1,2,4,8]$ and $\underline{y} = [1,3,8,8]$, $\underline{v} = [1,0,0,1]$.

8. Mapping: The mapping vector $\underline{x} = \mu(\underline{y} \longleftarrow \underline{z})$ is defined as follows:

$$x_i = \begin{cases} 0 \text{ if } \underline{u} = 0 \\ (\underline{u}/\underline{\nu})_1 \text{ if } \underline{u} \neq 0, \text{ where } \underline{\nu} = [1,2,3\ldots] \end{cases}$$

where $\underline{u} = (z_i \underline{\epsilon} = \underline{y})$. $\nu(\underline{x}) = \nu(\underline{z})$.

Each z_i is compared with all components of $\underline{y}$. If one or more of the components of $\underline{y}$ corresponds with z_i, $x_i = j$ where y_j is the first component of $\underline{y}$ that is equal to z_i. If there is no y_j that corresponds with z_i, $x_i = 0$. Since vectors can have repeated components, it is necessary to restrict the correspondent to the one occurring first. For example, if $\underline{y} = \left[\text{"A", "L", "A", "B", "A", "M", "A"}\right]$ and $\underline{z} = \left[\text{"L", "A", "B", "R", "A", "D", "O", "R"}\right]$, then $(\underline{y} \leftarrow \underline{z}) = \left[2,1,4,0,1,0,0,0\right]$ and $(\underline{z} \leftarrow \underline{y}) = \left[2,1,2,3,2,0,2\right]$.

The notion of a file (Φ) has been adopted to allow for the description of magnetic tape as a serial access memory. The operation of transferring an element from a file is reading ($x \leftarrow \Phi$), and the operation of transferring an element into a file is recording ($\Phi \leftarrow x$). A file can be read (or recorded) in the forward (denoted by ${}_0\Phi$) or backward (denoted by ${}_1\Phi$) direction, which will be indicated by the left subscripts. Right subscripts are used merely as indices. A file which is only recorded in an algorithm is an output file, and a file which is only read is an input file.

Both scalars and vectors can be read (or recorded) into files. In any step, the item to be read (or recorded) will serve to identify whether a scalar or vector is being read (or recorded). Thus, it is possible in one step to record a vector in the forward direction on a file, and in the next step, to read a scalar in the backward direction. In this event, the last element of the vector that was recorded in the file will be read out.

REFERENCES

1. Iverson, K. E., "The Description of Finite Sequential Processes," Theory of Switching, Report No. BL-23, Sec. III, Harvard Computation Laboratory, Cambridge, Mass. (1959).

2. Iverson, K. E. and Brooks, F. P. Jr., Automatic Data Processing, draft manuscript (to be published by Wiley, New York).

1. 1st Successor	$c \longleftarrow b\uparrow$	$c = b+1$	
2. 1st Predecessor	$c \longleftarrow b\downarrow$	$c = b-1$	
3. Weight	$k \longleftarrow \sigma(\underline{u})$	$k = \sum \underline{u}_j$	
4. Head vector	$\underline{u} \longleftarrow \underline{h}^j$	$u_i = 1 \longleftrightarrow i \leq j$	Dimension defined by context.
5. Tail vector	$\underline{u} \longleftarrow \underline{t}^j$	$u_{-j} = 1 \longleftrightarrow i \leq j$	
6. Full vector	$\underline{u} \longleftarrow \underline{\epsilon}$	$u_i = 1$	
7. Unit vector	$\underline{u} \longleftarrow \underline{\epsilon}^j$	$u_i = 1 \longleftrightarrow i = j$	
8. Identity permutation vector	$\underline{\nu}$	$\nu_k = k$	
9. Scalar replacement	$\underline{c} \longleftarrow a\underline{u}$	$c_i = au_i$	$\nu(\underline{c}) = \nu(\underline{u})$
10. Negation (not)	$\underline{w} \longleftarrow \bar{\underline{u}}$	$w_i = \bar{u}_i$	$\nu(\underline{w}) = \nu(\underline{u})$
11. Logical sum (or)	$\underline{w} \longleftarrow \underline{u} \vee \underline{v}$	$w_i = u_i \vee v_i$	$\nu(\underline{w}) = \nu(\underline{u}) = \nu(\underline{v})$
12. Logical product (and)	$\underline{w} \longleftarrow \underline{u} \wedge \underline{v}$	$w_i = u_i \wedge v_i$	$\nu(\underline{w}) = \nu(\underline{u}) = \nu(\underline{v})$
13. Vector compression	$\underline{c} \longleftarrow \underline{u}/\underline{a}$	$a_i \in \underline{c} \longleftrightarrow u_i = 1$	$\nu(\underline{a}) = \nu(\underline{u})$; $\nu(\underline{c}) = \sigma(\underline{u})$
14. Matrix row compression	$\underline{C} \longleftarrow \underline{u}/\underline{A}$	$\underline{C}^i = \underline{u}/\underline{A}^i$	$\nu(\underline{A}) = \nu(\underline{u})$; $\nu(\underline{C}) = \sigma(\underline{u})$; $\mu(\underline{C}) = \mu(\underline{A})$
14. Matrix column compression	$\underline{C} \longleftarrow \underline{u} // \underline{A}$	$\underline{C}_i = \underline{u}/\underline{A}_i$	$\mu(\underline{A}) = \nu(\underline{u})$; $\mu(\underline{C}) = \sigma(\underline{u})$; $\nu(\underline{C}) = \nu(\underline{A})$
15. Vector mesh	$\underline{z} \longleftarrow \backslash \underline{x}, \underline{u}, \underline{y} \backslash$	$\bar{\underline{u}}/\underline{z} = \underline{x}$, $\underline{u}/\underline{z} = \underline{y}$	$\nu(\underline{x}) = \sigma(\bar{\underline{u}})$; $\nu(\underline{y}) = \sigma(\underline{u})$; $\nu(\underline{z}) = \nu(\underline{u})$
16. Logical reduction "R"	$\underline{w} \longleftarrow (\underline{a}\ R\ \underline{b})$	$w_i = 1 \longleftrightarrow (a_i\ R\ b_i)$	$\nu(\underline{w}) = \nu(\underline{a}) = \nu(\underline{b})$
16a. Logical reduction "R" = "="	$\underline{w} \longleftarrow (\underline{a} = \underline{b})$	$w_i = 1 \longleftrightarrow (a_i = b_i)$	

Operations Used in Programs in This Report

TABLE A-1

17. Mapping	$\underline{c} \longleftarrow \mu(\underline{a} \longleftarrow \underline{b})$	$c_i = 0 \longleftrightarrow \underline{u} = 0$ $c_i = (\underline{u}/\underline{\nu})_i \longleftrightarrow \underline{u} \neq 0$ where $\underline{u} \longleftarrow (b_i \; \underline{\epsilon} \; = \underline{a})$	$\nu(\underline{c}) = \nu(\underline{b})$
18. Read from file (forward direction)	$x \longleftarrow {}_0\Phi$		
19. Record onto file (backward direction)	${}_1\Phi \longleftarrow x$		

TABLE A-1 (continued)

Appendix B

ERRATA SHEET FOR TABLES IN

1. Matejka, L., "Grammatical Specifications in the Russian-English Dictionary," Design and Operation of Digital Calculating Machinery, Report No. AF-50, Harvard Computation Laboratory, Section V (1958).

    a. p. V-21: In class N1.4, the "Gp" and "GpAp" listed for affix -ов should be listed instead for affix -ев.

    b. p. V-30: Line (11) should contain "GpApPp" wherever there is "GpAp".

    c. p. V-31: Line (7) should contain "NsAs" in column "M" of "A6+A7+A8" with an asterisk to a note stating that this refers to class A8 only.

    d. p. V-31: Line (25) should contain "GpApPp" wherever there is "GpAp".

2. Matejka, L., "The Automatic Interpretation of Russian Verbal Endings," Mathematical Linguistics and Automatic Translation, Report No. NSF-2, Harvard Computation Laboratory, Section III (1959).

    a. p. III-16:

        (1) The entry in "V4.02", "у", should be "Δ" instead of "C1s1".

        (2) The entry in "V4.1", "я", should be "B5" instead of "Δ".

        (3) The entry in "V4.11", "я", should be "B5" instead of "Δ".

    b. p. III-17: The entry in "V5.3", "мте", should be "Δ" instead of "C1p2".

    c. p. III-19:

(1) The entry in "V17", "ж", should include "B5" in addition to "B3p".

(2) The entry in "V19", "жте", should not be shaded in.

(3) The entry in "V15.1", "я", should be "B5" instead of "Δ".

(4) The entry in "V19", "а", should include "B5" in addition to "B3fs".

Appendix C

GENERATING AFFIX - AFFIX MAPPING FOR THE EXPERIMENTAL DICTIONARY

The first two columns of the tables in this appendix are reproduced from Magassy's original paradigm tables.[1] The generating affixes are listed for all but the canonical form.

The affixes that are factored in the place of each generating affix of each class (Sec..3.3) are listed in columns 3 and 4. If the stem remaining after the affix is factored is the same stem that remains after most or all of the affixes of the paradigm are factored, then the affix is listed in column 3. Otherwise, the affix is listed in column 4. The affixes in the fourth column are a result of false factoring (Sec. 3-2B).

No attempt has been made in these tables to include information on vowel insertion or vowel substitution in the generating stems. Reference for such information should be made to Oettinger's updated tables.[2]

If an affix appears twice in a class, each occurrence is identified with an asterisk, denoting the presence of a potential artificial affix homograph in the experimental dictionary.

REFERENCES

1. Magassy, K., "An Automatic Method of Inflection for Russian (II)," Design and Operation of Digital Calculating Machinery, AF-49, Sec. III, Progress Reports by the Staff of the Computation Laboratory of Harvard University to the United States Air Force, Cambridge, Massachusetts (1957).

2. Oettinger, A. G., Automatic Language Translation: Lexical and Technical Aspects, Harvard University Press, Cambridge, Massachusetts (1960).

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
N1	Word		-∅	-в -ев -ов* -ам* -ем -им -ом* -ым -ям -ат -ет -ит -ут -ют -ят
	Stem	-а -у -ом -е -ы -ов -ам -ами -ах	-а -у -ом* -е -ы -ов* -ам* -ами -ах	-ему -ому -ете -ите
N1.1	Word		-∅	-ах* -их -ых -ях
	Stem	-а -у -ом -е -и -ов -ам -ами -ах	-а -у -ом -е -и -ов -ам -ами -ах*	
N1.2	Word		-∅	-ев -ов* -ем -ом* -ет
	Stem	-а -у -ом -е -ы -ов -ам -ами -ах	-а -у -ом* -е -ы -ов* -ам -ами -ах	

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
N1.3	Word Stem	-а -у -ом -е -и -ов -ам -ами -ах	-∅ -а -у -ом -е -и -ов -ам -ами -ах	
N1.4	Word Stem	-а -у -ем -е -и -ев -ам -ами -ах	-∅ -а -у -ем -е -и -ев -ам -ами -ах	
N2	Word Stem	-я -ю -ем -е -и -ев -ям -ями -ях	-й -я -ю -ем -е -и -ев -ям -ями -ях	-ей -ий -ой -ая -ее -ие -ое

Generating Affix - Affix Mapping for the Noun Morphological Type Classes

TABLE C-1

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
N2.1	Word Stem -я -ю -ем -е -и -ев -ям -ями -ях	-ей -я -ем -е -и -ев -ям -ями -ях	-ью
N3	Word Stem -я -ю -ем -е -и -ей -ям -ями -ях	-ь -я -ю -ем -е -и -ей -ям -ями -ях	-сь -ть -ся -ете -ите
N3.05	Word Stem -я -ю -ем -и -ей -ям -ями -ях	-ь -я -ю -ем -и -ей -ям -ями -ях	-сь -ть -ся

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
N3.1	Word Stem -а -у -ом -е -и -ей -ам -ами -ах -ем	-# -а -у -ом -е -и -ей -ам -ами -ах -ем	
N4	Word Stem -ы -е -у -ой -ей -# -ам -ами -ах	-а -ы -е -у -ой -ей -# -ам* -ами -ах	-ете -ите -ему -ому -в -ев -ов -ам* -ем -им -ом -ым -ям -ат -ет -ит -ут -ят -ют
N4.05	Word Stem -ы -е -у -ой -# -ам -ами -ах	-а -ы -е -у -ой -# -ам -ами -ах	-ем

TABLE C-1 (continued)

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping	Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
N4.06	Word Stem -ы -е -у -ой -# -ам -ами -ах	-а -ы -е -у -ой -# -ам -ами -ах	-ом	N5	Word Stem -и -е -ю -ей -ь -ям -ями -ях	-я -и -е -ю -ей -ь -ям -ями* -ях	-ся -ами -ями* -ьм -ями -ете -ите -сь -ть
N4.1	Word Stem -и -е -у -ой -ей -# -ам -ами -ах	-а -и -е -у -ой -ей -# -ам -ами -ах*	-ами -ах* -ях -ьх -ях	N5.05	Word Stem -и -е -ю -ей -й -ям -ями -ях	-я -и -е -ю -ей* -й -ям -ями -ях	-ая -яя -ее -ое -ье -ую -ей* -ой -ьй
N4.3	Word Stem -и -е -у -ой -# -ам -ами -ах	-а -и -е -у -ой -# -ам -ами -ах		N5.1	Word Stem -и -е -ю -ей -ь -ям -ями -ях -#	-я -и -е -ю -ей -ь -ям -ями -ях -#	
N4.31	Word Stem -и -е -у -ой -# -ам -ами -ах	-а -и -е -у -ой -# -ам -ами -ах		N5.15	Word Stem -и -е -ю -ей -ь -ям -ями -ях	-я -и -е -ю -ей -ь -ям -ями -ях	

TABLE C-1 (continued)

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
N5.2	Word Stem -и -е -ю -ей -й -ям -ями -ях	-я -и -е -ей* -ям -ями -ях	-ью -ей*
N5.3	Word Stem -и -е -ю -ей -й -ям -ями -ях	-я -и -е -ей -й -ям -ями -ях	-ью -ий
N6	Word Stem -и -ью -ей -ям -ями -ьми -ях	-ь -и* -ью -ей -ям -ями -ях	-сь -ть -и*
N6.1	Word Stem -и -ью -ей -ам -ами -ьми -ах	-ь -и* -ью -ей -ам -ами -ах	-ешь -ишь -и*
N7	Word Stem -и -ю -ей -й -ям -ями -ях	-я -и -ю -ей -й -ям -ями -ях	-ий
N8	-а -у -ом -е -# -ам -ами -ах	-о -а -у -ом* -е -# -ам* -ами -ах*	-его -ого -ему -ому -ете -ите -в -ев -ов -ам* -ем -им -ом* -ым -ям -ат -ет -ит -ут -ют -ят -ах* -их -ых -ях
N8.1	Word Stem -а -у -ом -е -# -ам -ами -ах	-о -а -у -ом* -е -# -ам -ами -ах	-ом*

TABLE C-1 (continued)

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
N8.15	Word Stem	-а -у -ом -е -# -ам -ами -ах	-о -а -у -ом -е -# -ам -ами -ах	-ем -ям
N8.3	Word Stem	-а -ом -е -# -ам -ами -ах -и -ов	-о -а -ом -е -# -ам -ами -ах -и -ов	
N8.4	Word Stem	-а -у -ом -е -# -ам -ами -ах -и	-о -а -у -ом -е -# -ам -ами -ах -и	
N9	Word Stem	-а -у -ем -ев -# -ам -ами -ах	-е -а -у -ем -ев -# -ам -ами -ах	
N9.2	Word Stem	-а -у -ем -# -ам -ами -ах	-е -а -у -ем -# -ам -ами -ах	
N10	Word Stem	-я -ю -ем -и -й -ям -ями -ях	-е -я -ю -ем -и -й -ям -ями -ях	-ие -ий
N11	Word Stem	-я -ю -ем -й -ям -ями -ях	-е -я -ем -ям -ями -ях	-ью -ей
N11.1	Word Stem	-я -ю -ем -й -ям -ями -ях	-е -я -ем -й -ям -ями -ях	-ью -ий

TABLE C-1 (continued)

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping	Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
N11.20	Word Stem -я -ю -ем -ев -ям -ями -ях	-е -я -ем -ев -ям -ями -ях	 -ью				
N12	Word Stem -ени -енем -ена -ен -енам -енами -енах	-я -и -ем -а -# -ам -ами -ах					
N13	Word Stem -я -ю -ем -ей -ям -ями -ях	-е -я -ю -ем -ей -ям -ями -ях					

TABLE C-1 (continued)

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
A1	Word		-ый	
	Stem	-ного	-ого	
		-ному	-ому	
		-ным	-ым	
		-ном	-ом	
		-ная	-ая	
		-ной	-ой	
		-ную	-ую	
		-ное	-ое	
		-ные	-ые	
		-ных	-ых	
		-ными	-ыми	
		-#	-#	
		-ен	-#	
		-на	-а	
		-но	-о	
		-ны	-ы	
		-а	-а	
		-о	-о	
		-ы	-ы	
A2	Word		-ой*	
	Stem		-ый	
		-ого	-ого	
		-ому	-ому	
		-ым	-ым	
		-ом	-ом	
		-ая	-ая	
		-ои	-ои*	
		-ую	-ую	
		-ое	-ое	
		-ые	-ые	
		-ых	-ых	
		-ыми	-ыми	
		-#	-#	
		-а	-а	
		-о	-о	
		-ы	-ы	

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
A3	Word		-ой*	
	Stem	-ый	-ый	
		-ого	-ого	
		-ому	-ому	
		-ым	-ым*	
		-ом	-ом*	
		-ая	-ая	
		-ой	-ой*	
		-ую	-ую	
		-ое	-ое	
		-ые	-ые	
		-ых	-ых	
		-ыми	-ыми	
		-#	-#	-в -ев -ов
				-ам -ем -им
				-ом* -ым* -ям
				-ат -ет -ит
				-ют -ут -ят
		-а	-а	
		-о	-о	
		-ы	-ы	
A4	Word		-ий	
	Stem	-его	-его	
		-ему	-ему	
		-им	-им	
		-ем	-ем	
		-ая	-ая	
		-ей	-ей	
		-ую	-ую	
		-ее	-ее	
		-ие	-ие	
		-их	-их	
		-ими	-ими	
		-#	-#	
		-а	-а	
		-о	-о	
		-е	-е	
		-и	-и	-вши

Generating Affix - Affix Mapping for the Adjective Morphological Type Classes

TABLE C-2

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping	Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
A5	Word Stem		-ий	-й	A7	Word Stem		-ый	
		-его	-его				-ого	-ого	
		-ему	-ему				-ому	-ому	
		-им	-им				-ым	-ым	
		-ем	-ем				-ом	-ом	
		-яя	-яя				-ая	-ая	
		-ей	-ей				-ой	-ой	
		-юю	-юю				-ую	-ую	
		-ее	-ее				-ое	-ое	
		-ие	-ие	-е*			-ые	-ые	
		-их	-их				-ых	-ых	
		-ими	-ими				-ыми	-ыми	
		-ь	-ь				-#	-#	
		-я	-я				-а	-а	
		-е	-е*				-о	-о	
		-и	-и				-ы	-ы	
A6	Word Stem		-ий		A8	Word Stem		-ый	
		-ого	-ого				-ой	-ой**	
		-ому	-ому				-ого	-ого	
		-им	-им				-ому	-ому	
		-ом	-ом				-ым	-ым	
		-ая	-ая				-ом	-ом	
		-ой	-ой				-ая	-ая	
		-ую	-ую				-ой	-ой**	
		-ое	-ое				-ую	-ую	
		-ие	-ие				-ое	-ое	
		-их	-их				-ые	-ые	
		-ими	-ими				-ых	-ых*	
		-#	-#				-ыми	-ыми	
		-а	-а				-#	-#	-ах -ех*
		-о	-о						-ых -ях
		-и	-и				-а	-а	
							-о	-о	
							-ы	-ы	-ами

TABLE C-2 (continued)

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V1	Word	-ть	
	Stem -ю	-ю	
	-ешь	-ешь	
	-ет	-ет	
	-ем	-ем	
	-ете	-ете	
	-ют	-ют	
	-л		- #
	-ла		-а
	-ло		-о
	-ли		-и
	-й	-й	-ей
	-йте		-е
	-я	-я	-ая -яя
	-в	-в	-ев
	-вши	-вши	
V2	Word	-ть	
	Stem -у	-у	
	-ешь	-ешь	
	-ет	-ет	
	-ем	-ем	
	-ете	-ете	
	-ут	-ут	
	-ул		- #
	-ула		-а
	-уло		-о
	-ули		-и*
	-и	-и*	
	-ите	-ите	
	-ув		-в
	-увши		-вши

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V2.01	Word	-ть	
	Stem -у	-у	
	-ешь	-ешь	
	-ет	-ет	
	-ем	-ем	
	-ете	-ете	
	-ут	-ут	
	-ул		-#
	-ула		-а
	-уло		-о
	-ули		-и
	-ь	-ь	
	-ьте		-е
	-ув		-в
	-увши		-вши
V3	Word	-ть	
	Stem -ую	-ую	
	-уешь		-ешь
	-ует		-ет
	-уем		-ем
	-уете		-ете
	-уют		-ют
	-евал, -овал		- #
	-евала, -овала		-а
	-евало, -овало		-о
	-евали, -овали		-и
	-уй		-й
	-уйте		-е
	-уя		-я
	-евав, -овав		-в
	-евавши, -овавши		-вши

Generating Affix - Affix Mapping for the Verb Morphological Type Classes

TABLE C-3

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V4	Word Stem -ю } -у } -ишь -ит -им -ите -ят -ил -ила -ило -или -и -я -ив -ивши	-ть -ю -у -ишь -ит -им -ите -ят -и* -я	-ую -# -а -о -и* -ами -ими -ьми -ями -ая -ся -в -вши
V4.01	Word Stem -у } -ю } -ишь -ит -им -ите -ят -ил -ила -ило -или -ь -я -ив -ивши -ьте	-ть* -у -ю -ишь -ит -им -ите -ят -ь -я	-# -а -о -и -сь -ть* -ся -в -вши -е
V4.02	Word Stem -ю -ишь -ит -им -ите -ят -ил -ила -ило -или -й -я -ив -ивши -йте	-ть -ю -ишь -ит -им -ите -ят -й -я	-ую -# -а -о -и -ей -ой -ая -в -вши -е
V4.1	Word Stem -у -ишь -ит -им -ите -ят -ил -ила -ило -или -и -я -ив -ивши	-ть -у -ишь -ит -им -ите -ят -и* -я	-# -а -о -и* -в -вши

TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm	
		Direct Mapping	Indirect Mapping
V4.11	Word Stem -у -ишь -ит -им -ите -ят -ил -ила -ило -или -ь -я -ив -ивши -ьте	-ть* -у -ишь -ит -им -ите -ят -ь -я	-# -а -о -и -ть* -в -вши -е
V4.2	Word Stem -у -ишь -ит -им -ите -ат -ил -ила -ило -или -и -а -ив -ивши	-ть -у -ишь -ит -им -ите -ат -и* -а*	-# -а* -о -и* -вши* -в -вши*
V4.21	Word Stem -у -ишь -ит -им -ите -ат -ил -ила -ило -или -ь -а -ив -ивши -ьте	-ть -у -ишь* -ит -им -ите -ат -ь -а*	-# -а* -о -и -ешь -ишь* -в -вши -е
V5	Word Stem -у -ешь -ет -ем -ете -ут -ал -ала -ало -али -и -ите -ь -ьте -я -а -ав -авши	-ть -у -ешь* -ет -ем -ете -ут -и* -ите -ь -я -а*	-# -а* -о -и* -вши* -ешь* -ишь -е -в -вши*

TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V5.1	Word		-ть	
	Stem	-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-ал		-#
		-ала		-а*
		-ало		-о
		-али		-и*
		-и	-и*	
		-ите	-ите	
		-ь	-ь	
		-ьте		-э
		-а	-а*	
		-ав		-в
		-авши		-вши
		-я	-я	
V5.2	Word		-ть	
	Stem	-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-ал		-#
		-ала		-а
		-ало		-о
		-али		-и*
		-и	-и*	-ами -ими
				-ьми -ями
		-ите	-ите	
		-ав		-в
		-авши		-вши

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V5.3	Word		-ть	
	Stem	-ю	-ю	-ую
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ют	-ют	
		-ял		-#
		-яла		-а
		-яло		-о
		-яли		-и
		-й	-й	-ей -ий
				-ой -ый
		-йте		-е
		-я	-я	-ая -яя
		-яв		-в
		-явши		-вши
V5.4	Word		-ть	
	Stem	-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-ал		-#
		-ала		-а
		-ало		-о
		-али		-и*
		-и	-и*	
		-ите	-ите	
		-я	-я	
		-ав		-в
		-авши		-вши

TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping	Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V5.41	Word Stem -у -ешь -ет -ем -ете -ут -ал -ала -ало -али -и -ите -я -ав -авши	-ть -у -ешь -ет -ем -ете -ут -и* -ите -я	-# -а -о -и* -в -вши	V6.1	Word Stem -ю -ишь -ит -им -ите -ят -ял -яла -яло -яли -й -йте -я -явши	-ть -ю -ишь -ит -им -ите -ят -я	-# -а -о -и -ой -е -вши
V6	Word Stem -у -ишь -ит -им -ите -ат -ал -ала -ало -али -ь -ьте -и -а -ав -авши	-ть -у -ишь -ит -им -ите -ат -ь -и* -а*	-# -а* -о -и* -е -в -вши	V6.2	Word Stem -у } -ю } -ишь -ит -им -ите -ят -ел -ела -ело -ели -и -ь -ьте -я -ев	-ть* -у -ю -ишь -ит -им -ите -ят -и* -ь -я -ев	-# -а -о -и* -ами -ями -ами -ями -сь -ть* -е -ся

TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm	
		Direct Mapping	Indirect Mapping
V7	Word		-ть
	Stem -ну	-у	
	-нешь	-ешь	
	-нет	-ет	
	-нем	-ем	
	-нете	-ете	
	-нут	-ут	
	-#		Any affix
	-ла		-а
	-ло		-о
	-ли		-и*
	-ни	-и*	
	-ните	-ите	
	-нь	-ь	
	-ньте		-е
	-нув		-в
	-нувши		-вши
	-ши		-и*
V8	Word	-и*	-ть
	Stem -у	-у	
	-ешь	-ешь	
	-ет	-ет	
	-ем	-ем	
	-ете	-ете	
	-ут	-ут	
	-#	-#	
	-ла		-а
	-ло		-о
	-ли		-и*
	-и	-и*	
	-ите	-ите	
	-ь	-ь	-сь
	-ьте		-е
	-я	-я	-ся
	-ши		-и*

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm	
		Direct Mapping	Indirect Mapping
V8.1	Word	-и*	-ть
	Stem -ту	-у	
	-тешь	-ешь	
	-тет	-ет	
	-тем	-ем	
	-тете	-ете	
	-тут	-ут	
	-л		-#
	-ла		-а
	-ло		-о
	-ли		-и*
	-ти	-и*	
	-тите	-ите	
	-тя	-я	
	-тши		-и*
V8.11	Word		-ть
	Stem -ту	-у	
	-тешь	-ешь	
	-тет	-ет	
	-тем	-ем	
	-тете	-ете	
	-тут	-ут	
	-л		-#
	-ла		-а
	-ло		-о
	-ли		-и*
	-ти	-и*	
	-тите	-ите	
	-тя	-я	
	-тши		-и*

TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V8.2	Word			-и* -ть
	Stem	-жу	-у	
		-дешь	-ешь	
		-дет	-ет	
		-дем	-ем	
		-дете	-ете	
		-дут	-ут	
		-л		-#
		-ла		-а
		-ло		-о
		-ли		-и*
		-ди	-и*	
		-дите	-ите	
		-дя	-я	
		-дши		-и*
V9	Word			-ь
	Stem	-гу	-у	
		-жешь		-ешь
		-жет		-ет
		-жем		-ем
		-жете		-ете
		-гут	-ут	
		-г	-#	
		-гла		-а*
		-гло		-о
		-гли		-и*
		-ги	-и*	
		-гите	-ите	
		-жа		-а*
		-гши		-и*

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V9.1	Word		-ь	
	Stem	-ку		-у
		-чешь	-ешь	
		-чет	-ет	
		-чем	-ем	
		-чете	-ете	
		-кут		-ут
		-к		-#
		-кла		-а*
		-кло		-о
		-кли		-и*
		-ки		-и*
		-ките		-ите
		-ча	-а*	
		-кши		-и*
V10	Word		-ть	
	Stem	-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-л		-#
		-ла		-а
		-ло		-о
		-ли		-и*
		-и	-и*	
		-ите	-ите	
		-в	-в	
		-вши	-вши	

TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V10.01	Word		-ть	
	Stem	-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-л		-#
		-ла		-а
		-ло		-о
		-ли		-и*
		-и	-и*	
		-ите	-ите	
		-в	-в	
		-вши	-вши	
V10.1	Word		-ть	
	Stem	-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-л		-#
		-ла		-а
		-ло		-о
		-ли		-и
		-и		-ими
		-ите	-ите	
		-в	-в	
		-вши	-вши	

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V10.2	Word		-ть	
	Stem	-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-л		-#
		-ла		-а
		-ло		-о
		-ли		-и*
		-и	-и*	
		-ите	-ите	
		-в	-в	
V10.3		возьму	-у	
		возьмешь	-ешь	
		возьмет	-ет	
		возьмем	-ем	
		возьмете	-ете	
		возьмут	-ут	
		взял		-#
		взяла		-а
		взяло		-о
		взяли		-и*
		возьми	-и*	
		возьмите	-ите	
		взяв		-в
		взявши		-вши
		взять		-ть

TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V10.4	Word	-ть	
	Stem -у	-у	
	-ешь	-ешь	
	-ет	-ет	
	-ем	-ем	
	-ете	-ете	
	-ут	-ут	
	-л		-#
	-ла		-а
	-ло		-о
	-ли		-и
	-и		-ими
	-ите	-ите	
	-в	-в	
	-вши	-вши	
V11	Word	-ть	
	Stem -у	-у	
	-ешь	-ешь	
	-ет	-ет	
	-ем	-ем	
	-ете	-ете	
	-ут	-ут	
	-#	-#	
	-ла		-а
	-ло		-о
	-ли		-и*
	-и	-и*	
	-ите	-ите	
	-ши		-и*

Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V11.1	Word	-ть	
	Stem -ю	-ю	
	-ешь	-ешь	
	-ет	-ет	
	-ем	-ем	
	-ете	-ете	
	-ют	-ют	
	-ол		-#
	-ола		-а
	-оло		-о
	-оли		-и*
	-и	-и*	
	-ите	-ите	
	-я	-я	
	-ов	-ов	
	-овши		-вши
V12	Word	-ть	
	Stem -ю	-ю	
	-ешь	-ешь	
	-ет	-ет	
	-ем	-ем	
	-ете	-ете	
	-ют	-ют	
	-вал		-#
	-вала		-а
	-вало		-о
	-вали		-и
	-вай		-й
	-вайте		-е
	-вая		-ая
	-вав		-в

TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V13	Word Stem		-ть	
		-ю		-ью
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ют	-ют	
		-л		- #
		-ла		-а
		-ло		-о
		-ли		-и
		-й		-ей
		-йте		-е
		-я	-я	
		-в	-в	
		-вши	-вши	
V14	Word Stem		-ть	
		-ю	-ю	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ют	-ют	
		-л		- #
		-ла		-а
		-ло		-о
		-ли		-и
		-й		-ой
		-йте		-е
		-я	-я	
		-в	-в	-ев
		-вши	-вши	

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V15	Word Stem		-ть	
		-ю		-ую
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ют	-ют	
		-л		- #
		-ла		-а
		-ло		-о
		-ли		-и
		-й	-й	
		-йте		-е
		-я	-я	
		-в	-в	
		-вши	-вши	
V15.1	Word Stem		-ть	
		-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-л		- #
		-ла		-а
		-ло		-о
		-ли		-и*
		-и	-и *	
		-ите	-ите	
		-в	-в	
		-вши	-вши	
		-я	-я	

TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V15.2	Word Stem		-ть	
		-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-л		-#
		-ла		-а
		-ло		-о
		-ли		-и
		-ь	-ь	
		-ьте		-е
		-в	-в	-ев
		-вши	-вши	
V16	Word Stem		-и*	
		-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-л		-#
		-ла		-а
		-ло		-о
		-ли		-и*
		-и	-и*	
		-ите	-ите	
		-учи		-и*
		-я	-я	
		-дши		-и*

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V17	Word Stem		-ть	
		-у	-у	
		-ешь	-ешь	
		-ет	-ет	
		-ем	-ем	
		-ете	-ете	
		-ут	-ут	
		-ал		-#
		-ала		-а
		-ало		-о
		-али		-и*
		-учи		-и*
		-ав		-в
		-авши		-вши
V18	Word Stem		-ть	
		-м		-ам
		-шь		-ь
		-от		-#*
		-дим		-им
		-дите		-ите
		-дут		-ут
		-л		-#*
		-ла		-а
		-ло		-о
		-ли		-и
		-й	-й	
		-йте		-е
		-в	-в	
		-вши	-вши	

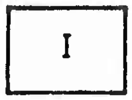
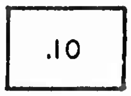
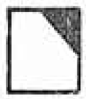
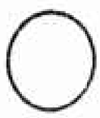
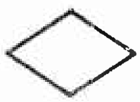
TABLE C-3 (continued)

Class Marker	Reduced Paradigm with Generating Affixes		Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping	Class Marker	Reduced Paradigm with Generating Affixes	Affixes Potentially Factional Inverse Inflection Algorithm: Direct Mapping	Affixes Potentially Factional Inverse Inflection Algorithm: Indirect Mapping
V19	Word		-ть		V21	быть		-ть*
	Stem	-у	-у			есть		-ть*
		-ишь	-ишь			суть		-ть*
		-ит	-ит			буду	-у	
		-им	-им			будешь	-ешь	
		-ите	-ите*			будет	-ет	
		-ут	-ут			будем	-ем	
		-ал		-#		будете	-ете	
		-ала		-а*		будут	-ут	
		-ало		-о		был		-#
		-али		-и*		была		-а
		-и	-и*			было		-о
		-ите	-ите*			были		-и*
		-а	-а*			будь	-ь	
		-ав		-в		будьте		-е
						будучи		-и*
V20	Word		-ть			бывши		-вши
	Stem	-у	-у					
		-ешь	-ешь					
		-ет	-ет					
		-им	-им					
		-ите	-ите					
		-ят	-ят					
		-ел		-#				
		-ела		-а				
		-ело		-о				
		-ели		-и*				
		-и	-и*					
		-ев	-ев					

TABLE C-3 (continued)

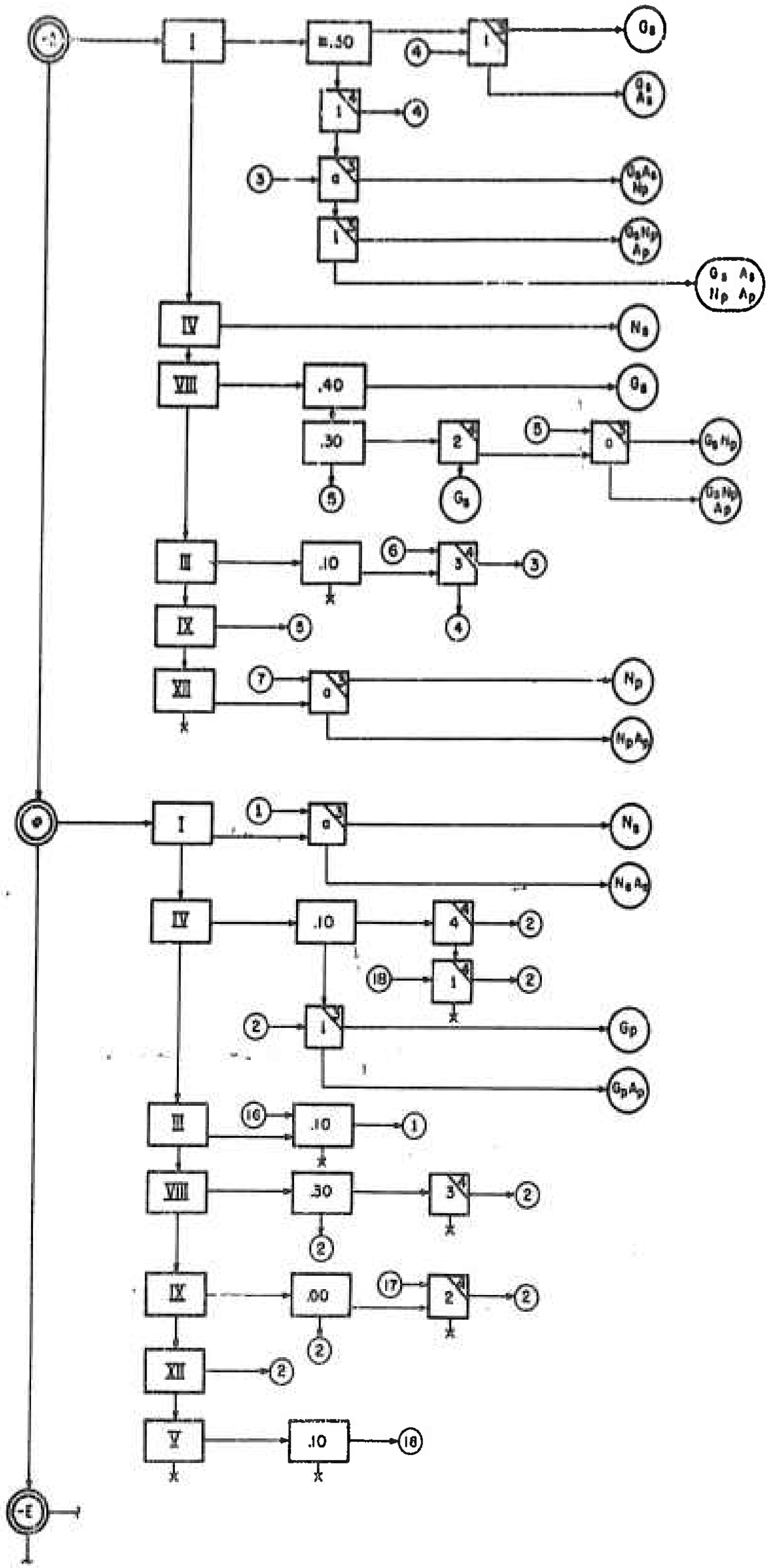
Appendix D

FLOW CHARTS FOR ANALYZER PROGRAMS

	RUSSIAN AFFIX
	CLASS MARKER (LEFT OF DECIMAL POINT)
	CLASS MARKER (RIGHT OF DECIMAL POINT)
	n th CHARACTER OF ORGANIZED WORD
	3rd SEMI-ORGANIZED WORD
	TREE TERMINAL WITH LEXICAL ATTRIBUTES
	CONNECTOR TO ANOTHER PAGE OF CHART
	ANOMALOUS STEM
	POSITIVE IDENTIFICATION AT BRANCH
	NO IDENTIFICATION AT BRANCH
X	INCOMPATIBLE TREE TERMINAL

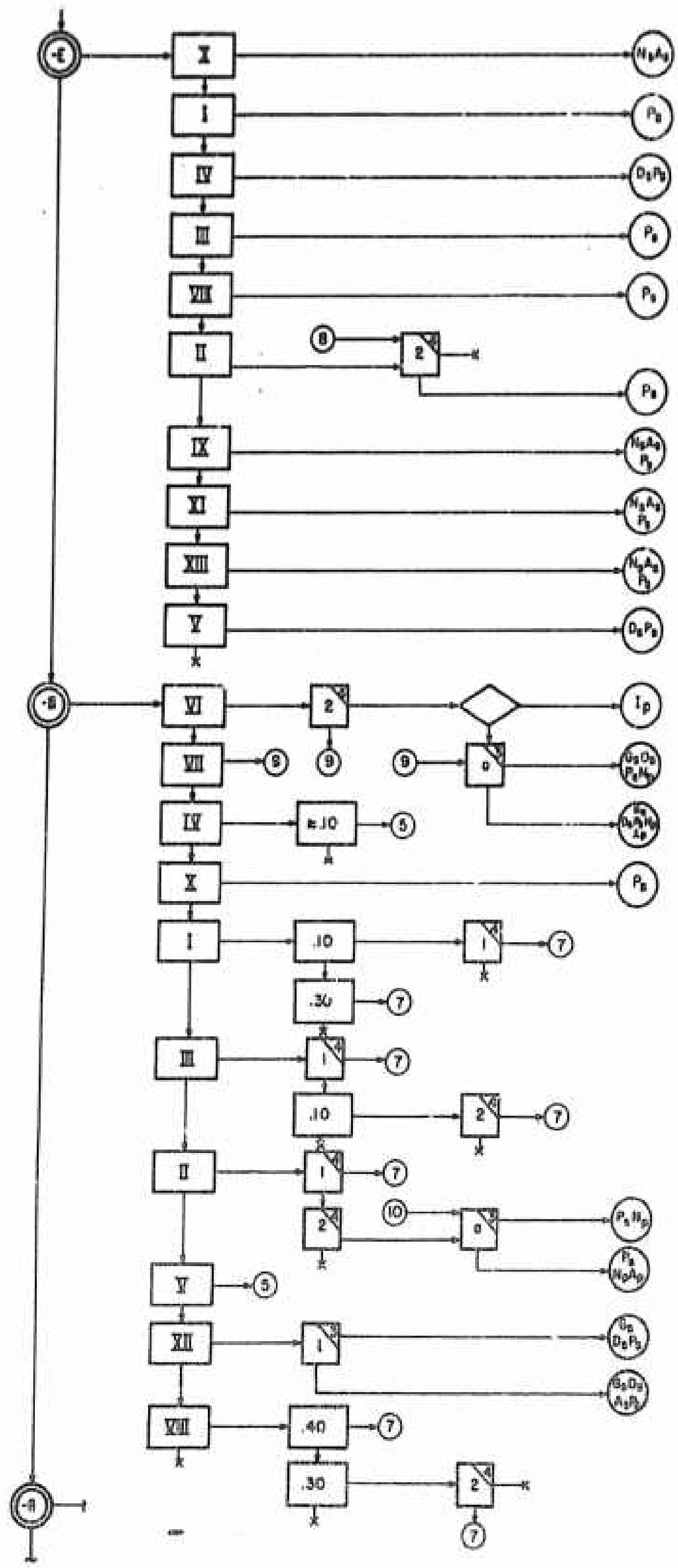
Key to Flow Charts

TABLE D-1



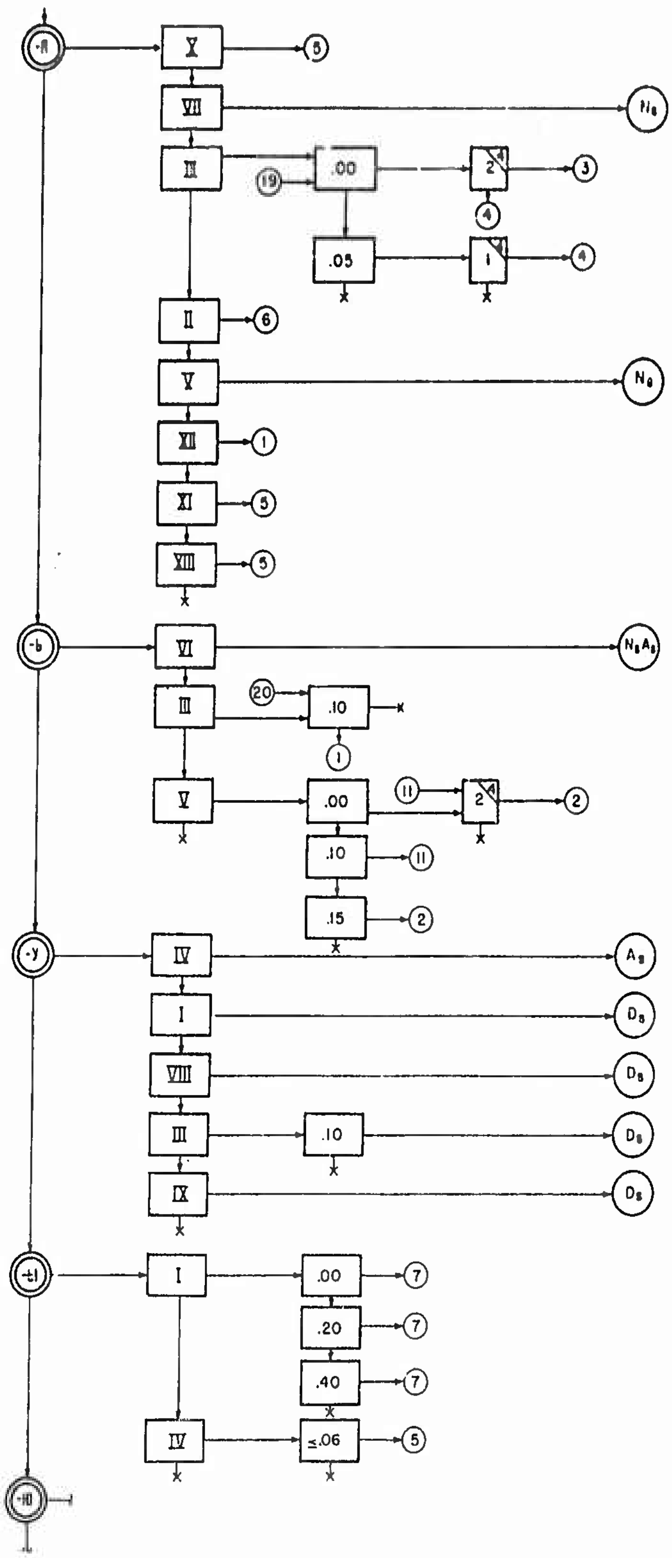
Noun Analyzer Program
Chart No. 1

Flow Chart D-1



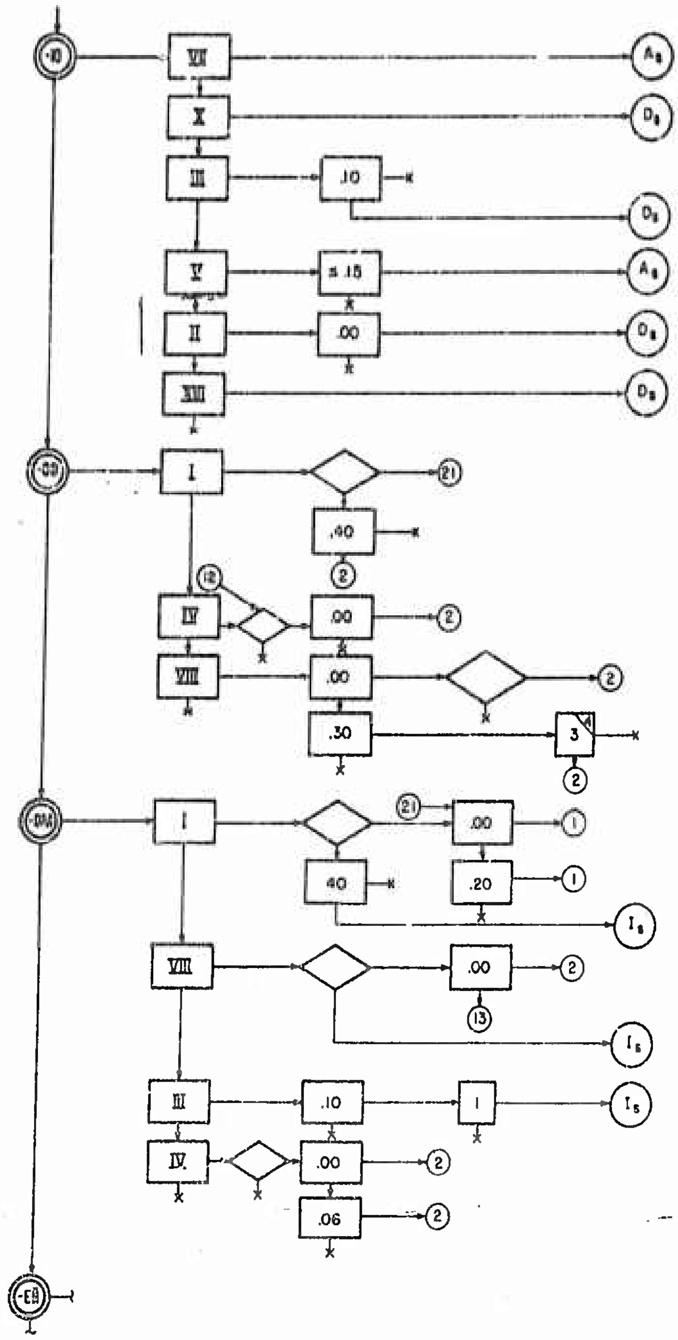
CONNECTOR NO.	TO CHART NO.
5	1
7	1

Flow Chart D-1 (continued)
Chart No. 2



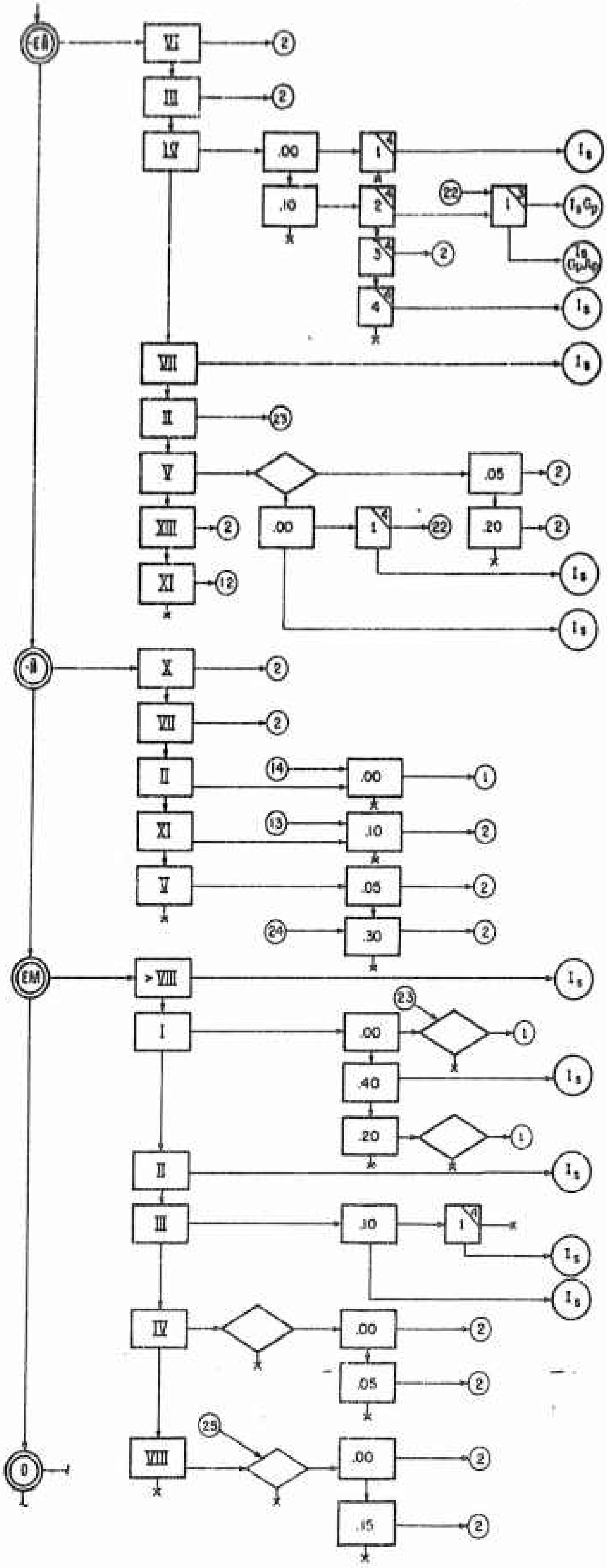
CONNECTOR NO.	TO CHART NO.
1	1
2	1
3	1
4	1
5	1
6	1
7	1

Flow Chart D-1 (continued)
Chart No. 3



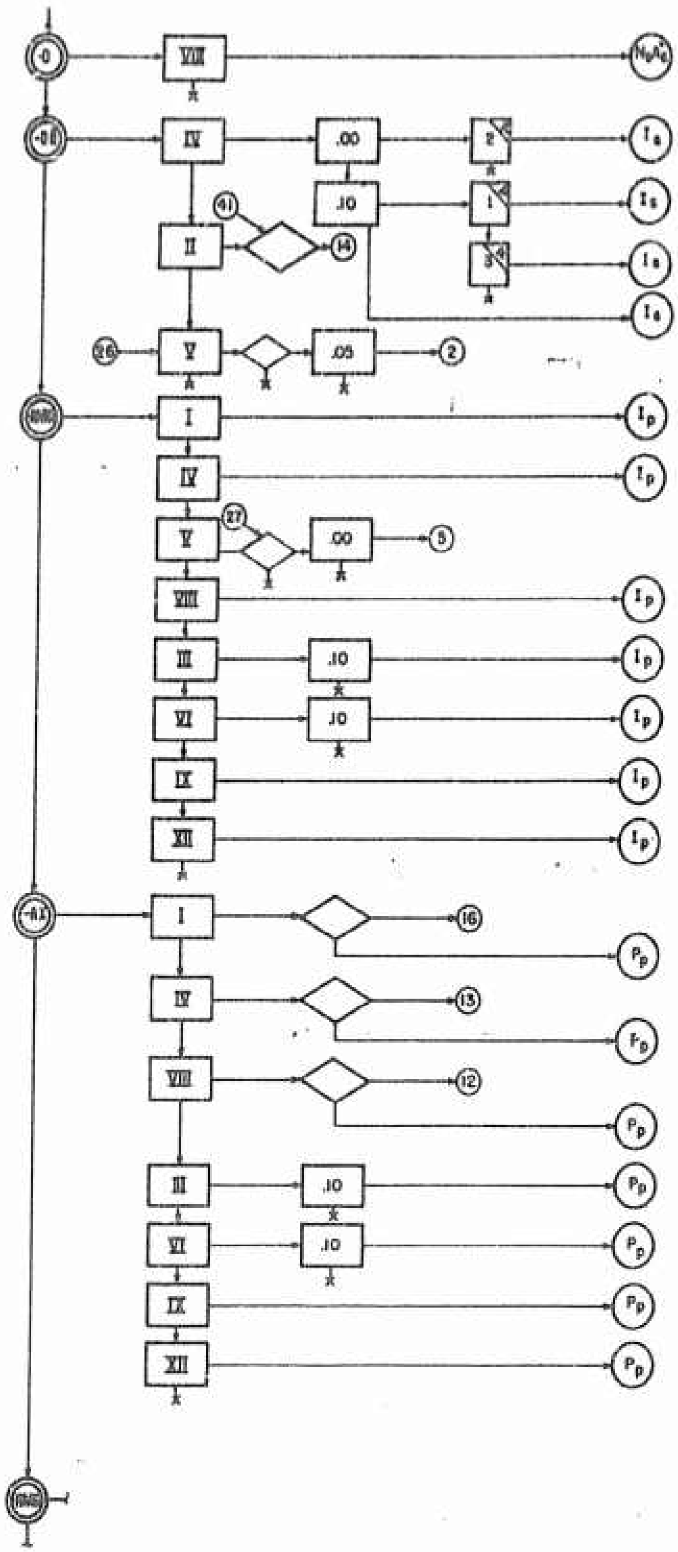
CONNECTOR NO.	TO CHART NO.
1	1
2	1
13	5

Flow Chart D-1 (continued)
Chart No. 4



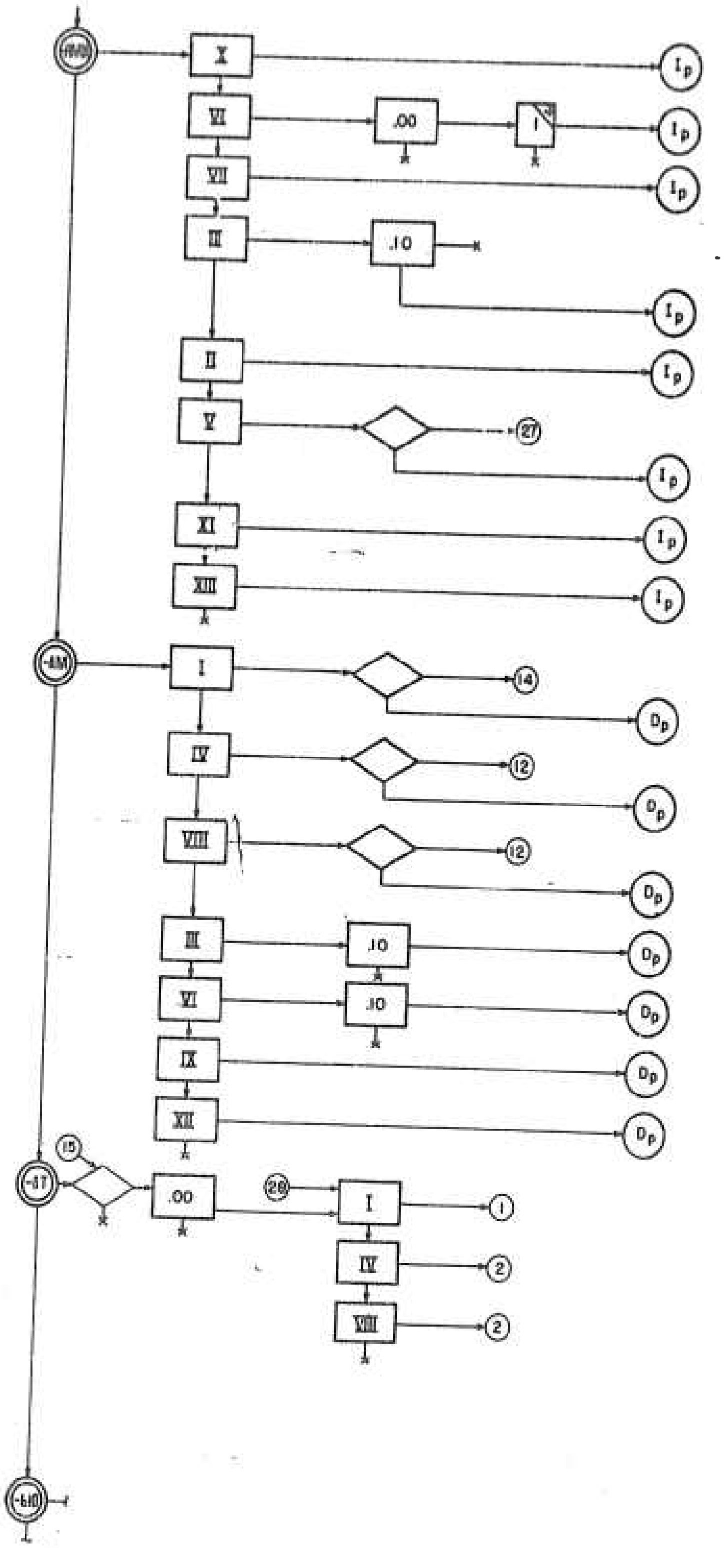
CONNECTOR NO.	TO CHART NO.
1	1
2	1
12	4

Flow Chart D-1 (continued)
Chart No. 5



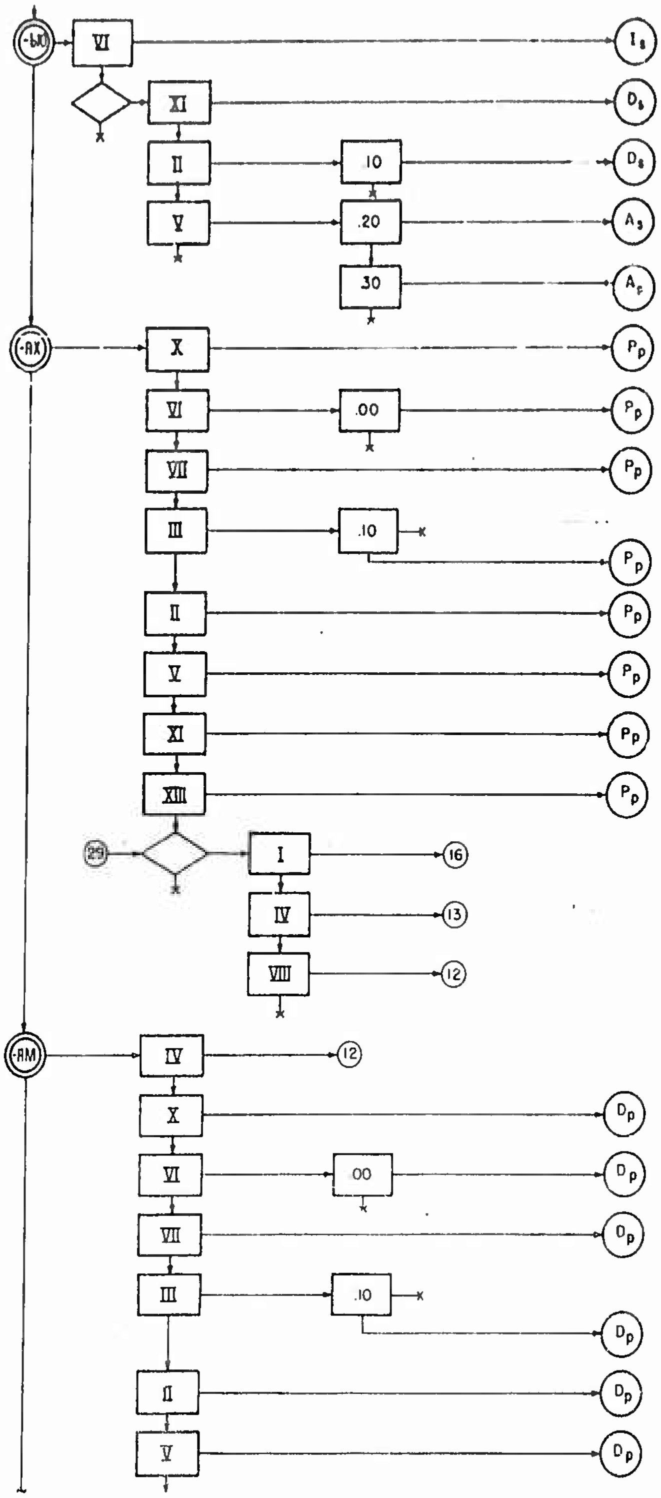
CONNECTOR NO.	TO CHART NO.
2	1
5	1
12	4
13	5
14	5
16	1

Flow Chart D-1 (continued)
Chart No. 6



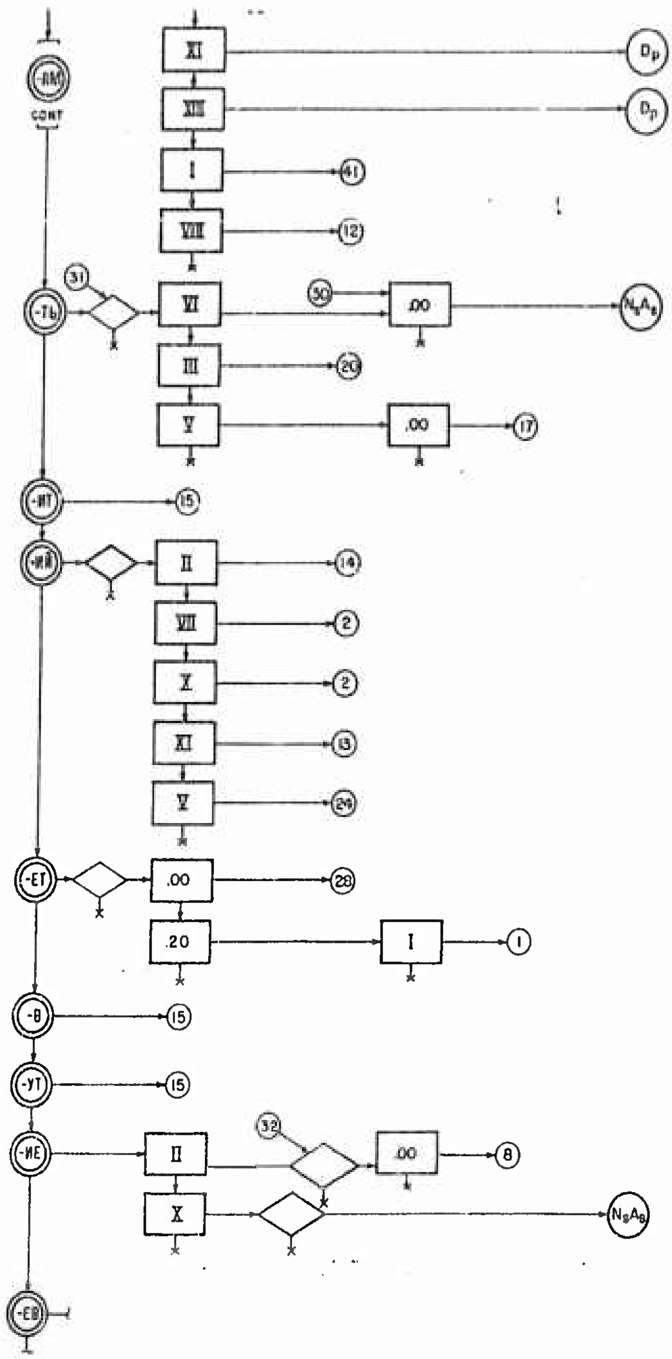
CONNECTOR NO.	TO CHART NO.
1	1
2	1
12	4
14	5
27	6

Flow Chart D-1 (continued)
Chart No. 7



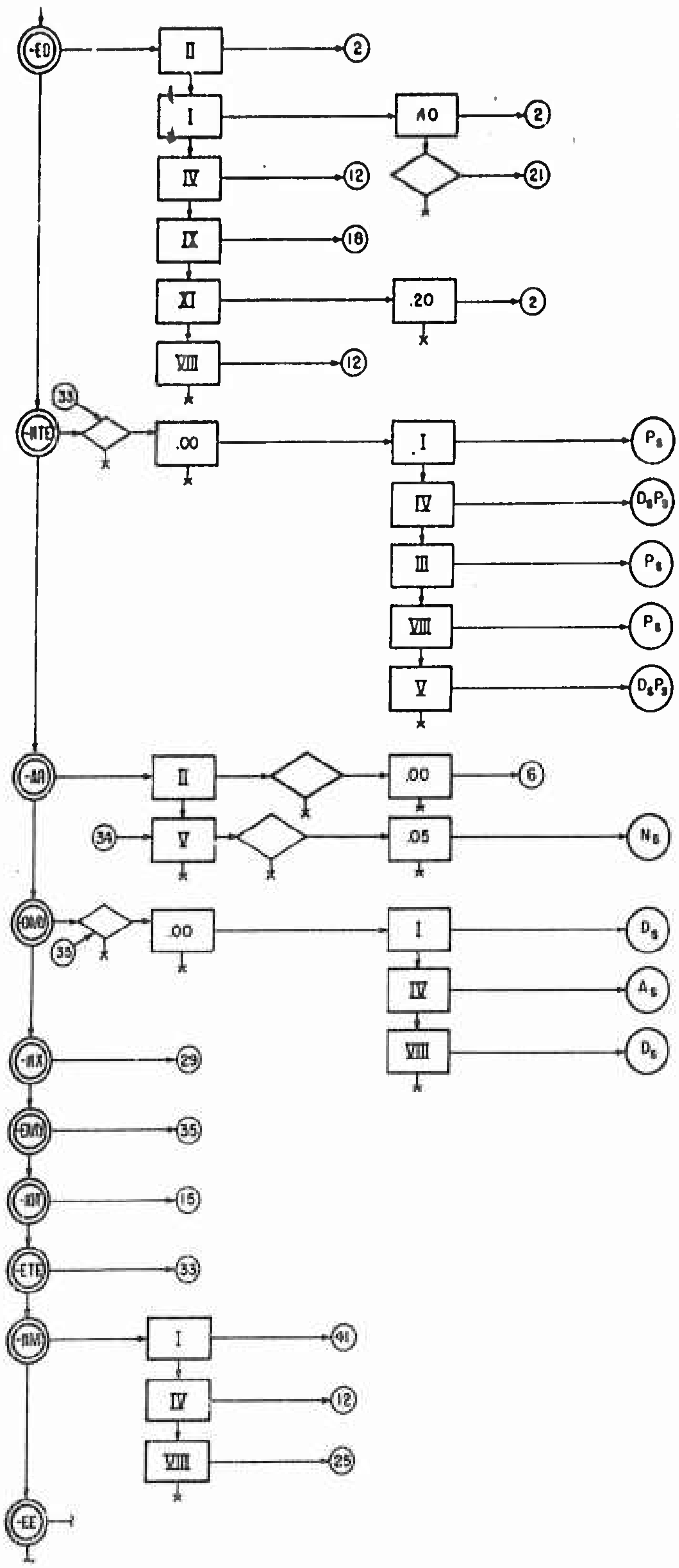
CONNECTOR NO	TO CHART NO
12	4
13	5
16	1

Flow Chart D-1 (continued)
Chart No. 8



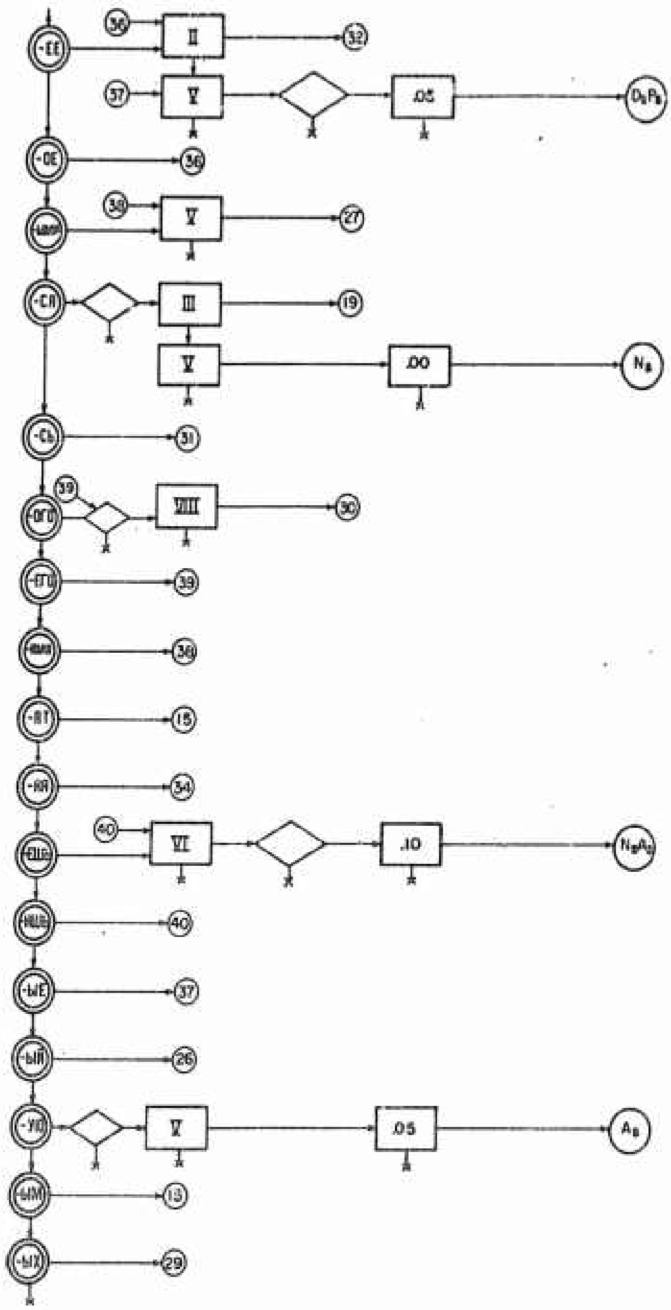
CONNECTOR NO.	TO CHART NO.
1	1
2	1
8	2
12	4
13	5
14	5
15	7
17	1
20	3
24	5
28	7
41	6

Flow Chart D-1 (continued)
Chart No. 9



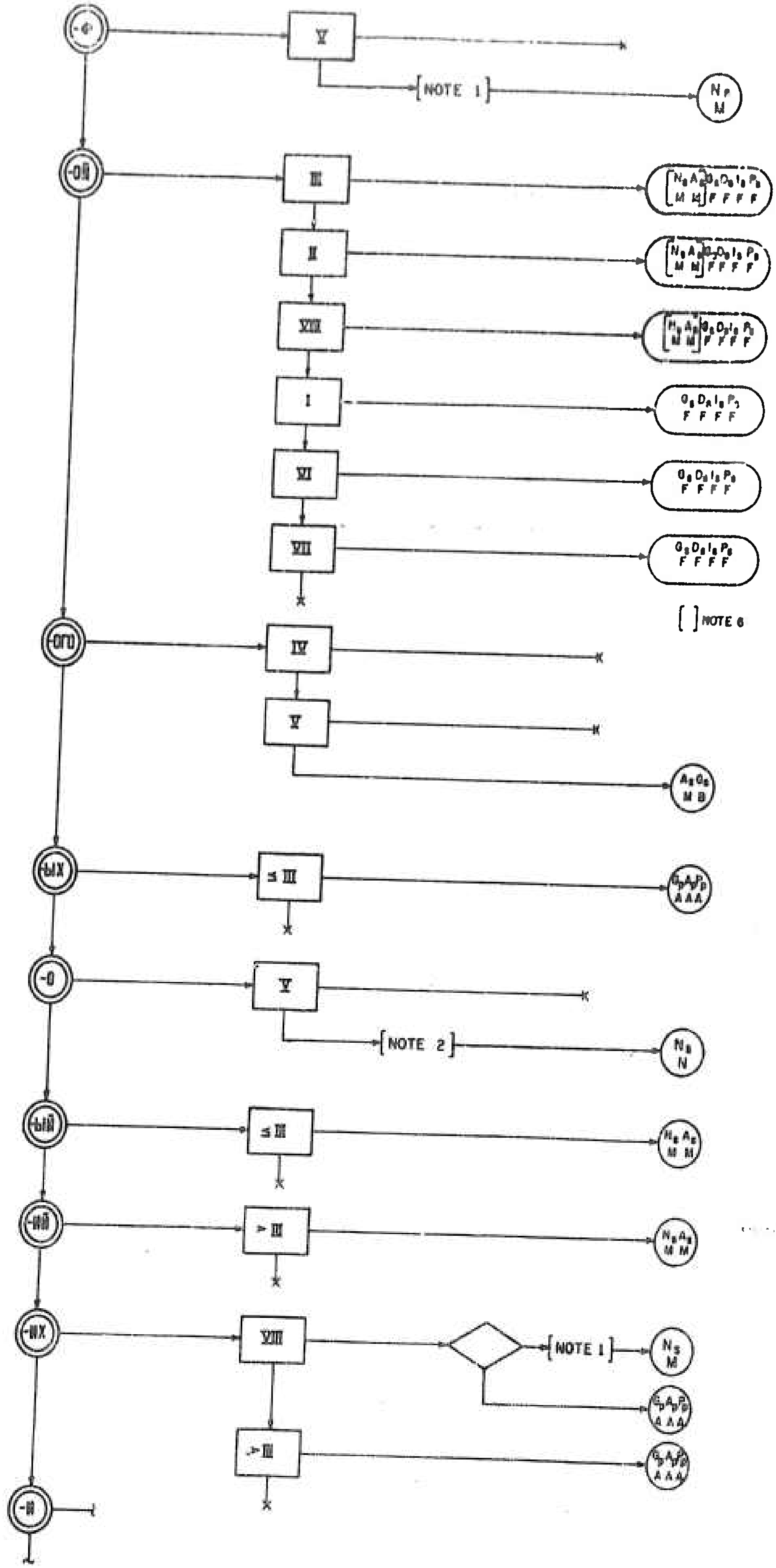
CONNECTOR NO.	TO CHART NO.
2	1
6	1
12	4
14	5
15	7
18	1
21	4
25	5
29	8
41	6

Flow Chart D-1 (continued)
Chart No. 10



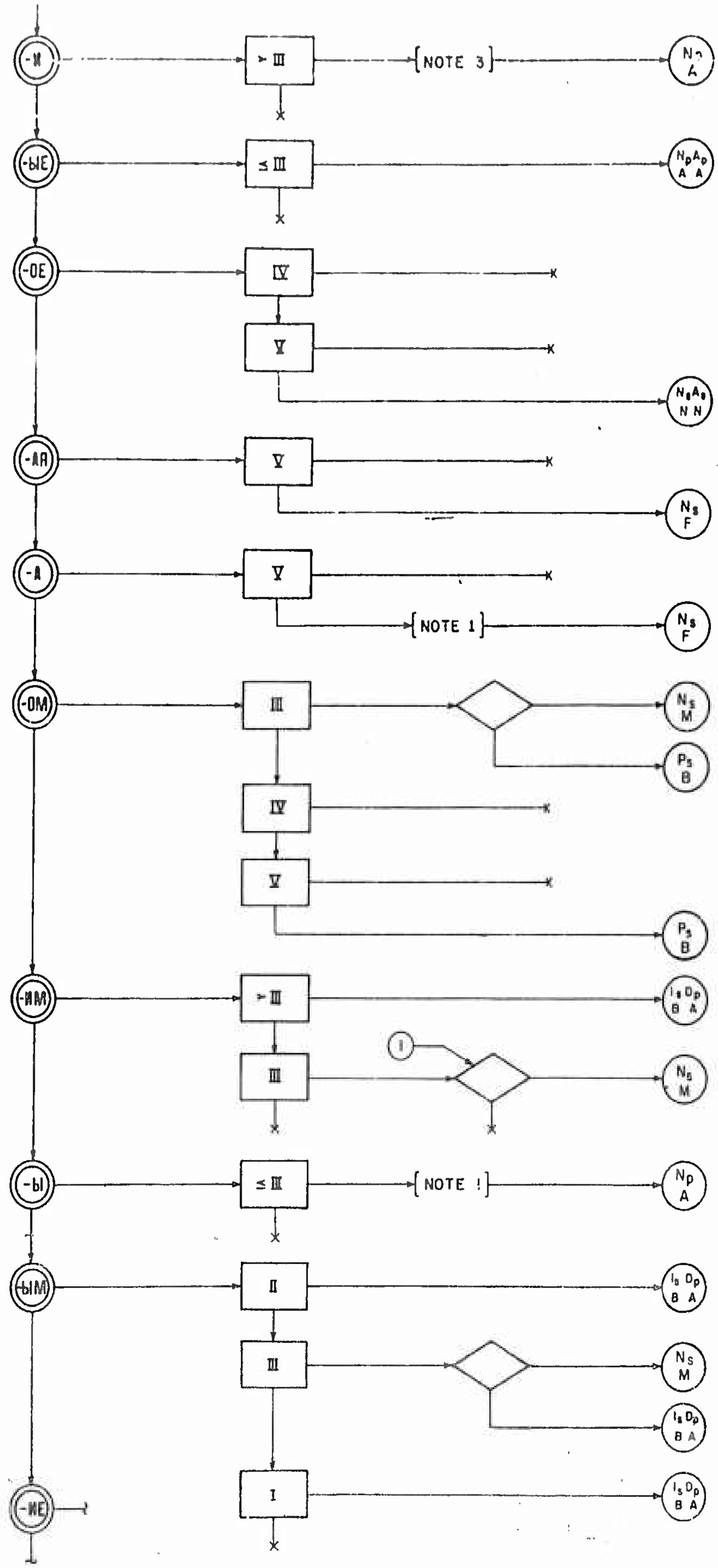
CONNECTOR NO.	TO CHART NO.
15	7
19	3
26	6
27	6
29	8
30	9
31	9
32	9
34	10

Flow Chart D-1 (continued)
Chart No. 11

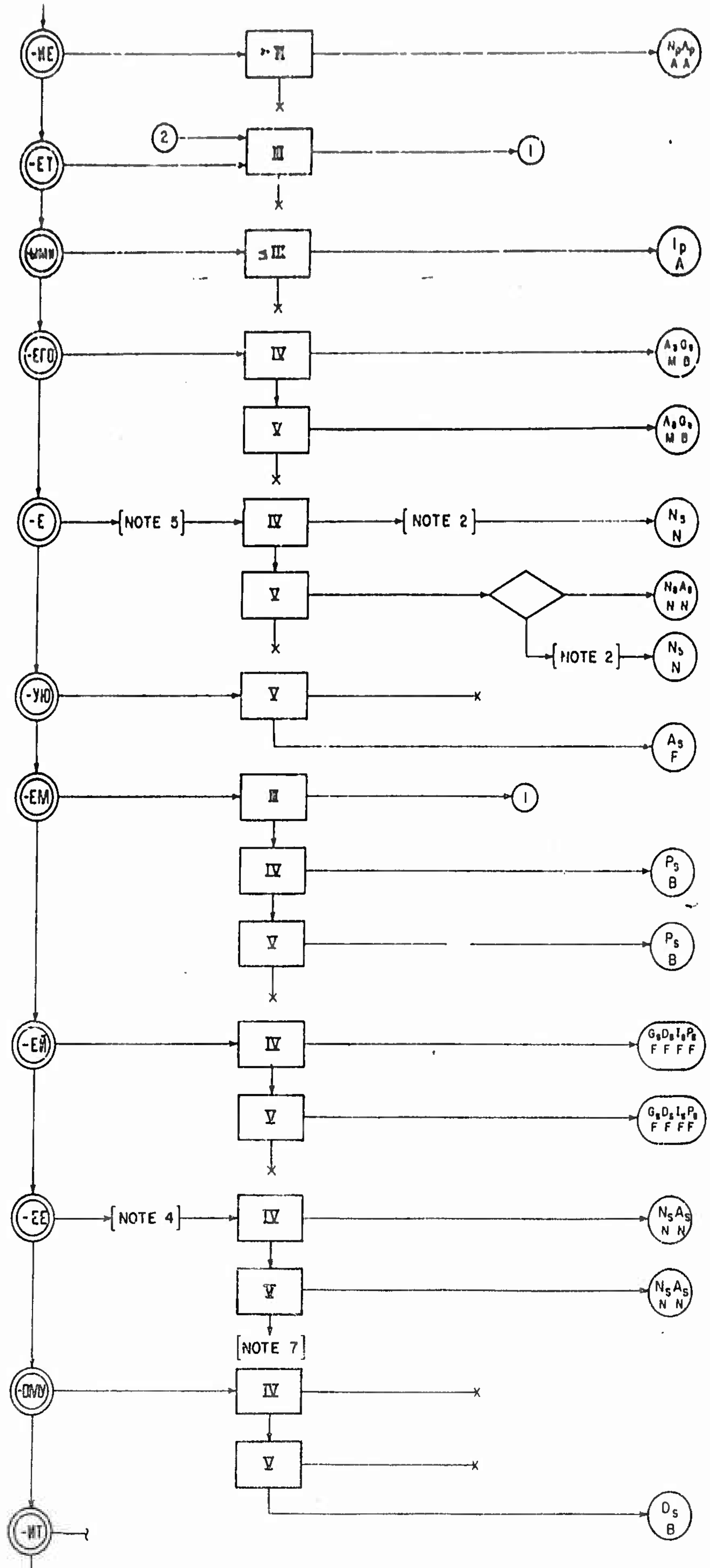


Adjective Analyzer Program
Chart No. 1

Flow Chart D-2

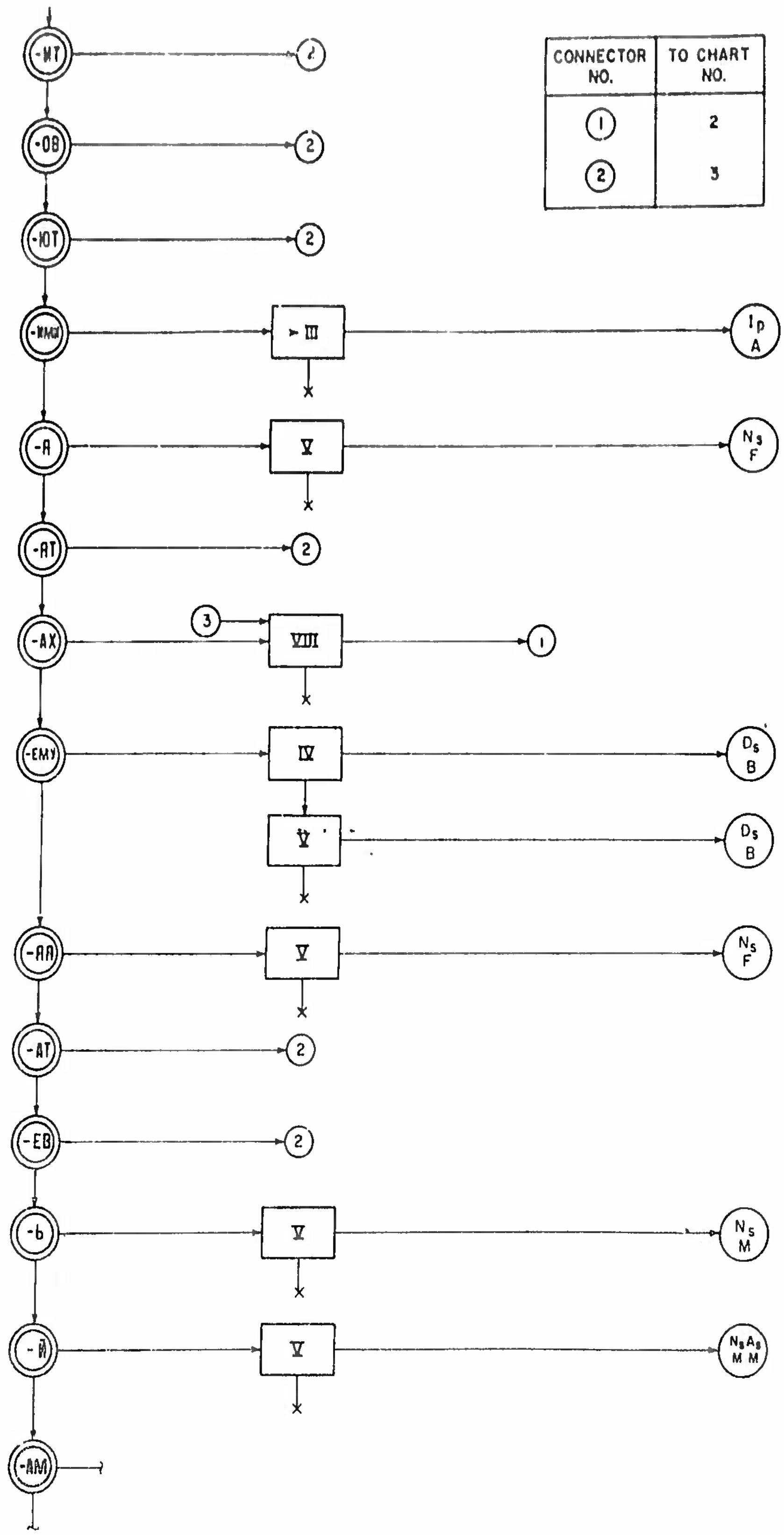


Flow Chart D-2 (continued)
Chart No. 2

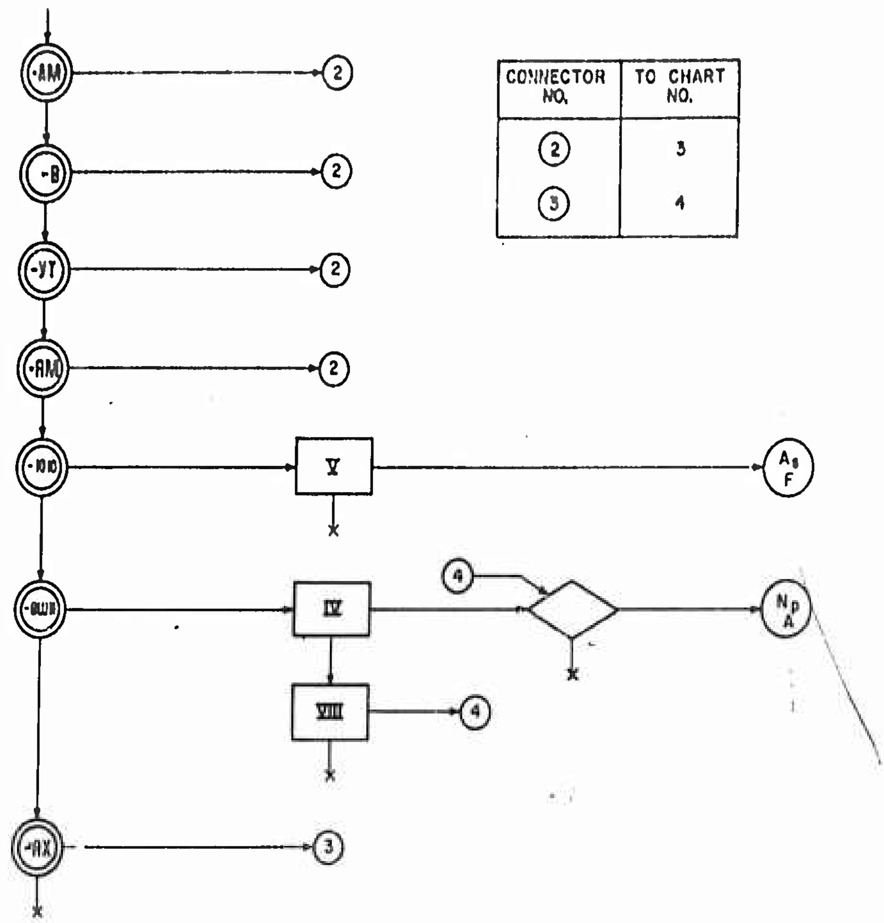


CONNECTOR NO.	TO CHART NO.
1	2

Flow Chart D-2 (continued)
Chart No. 3



Flow Chart D-2 (continued)
Chart No. 4



CONNECTOR NO.	TO CHART NO.
2	3
3	4

Flow Chart D-2 (continued)
Chart No. 5

Note 1: Insert a "1" into character position 8 of the organized word.

Note 2: Insert a "2" into character position 8 of the organized word.

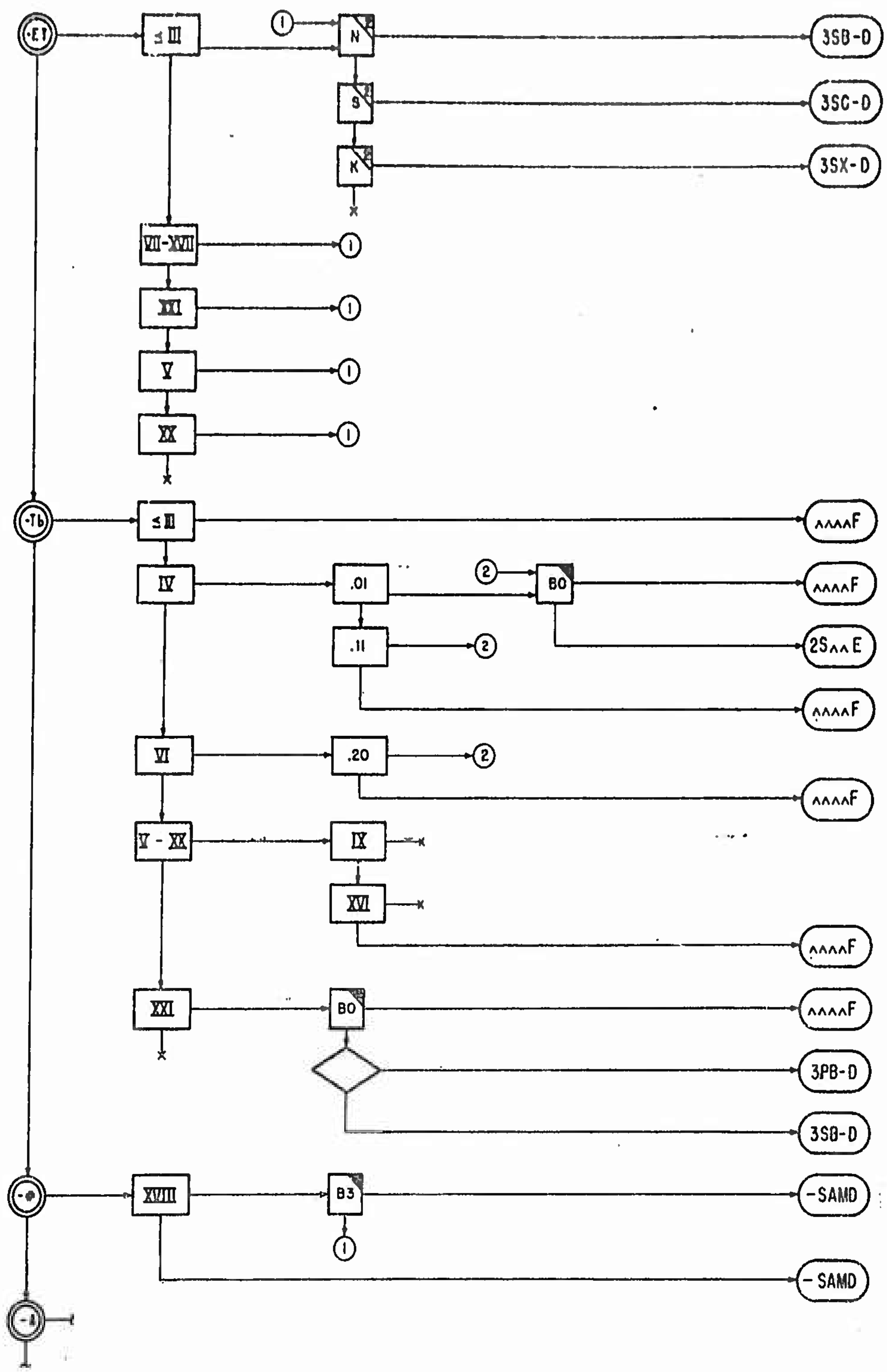
Note 3: Insert a "3" into character position 8 of the organized word.

Note 4: Insert a "1" into character position 9 of the organized word.

Note 5: Insert a "2" into character position 9 of the organized word.

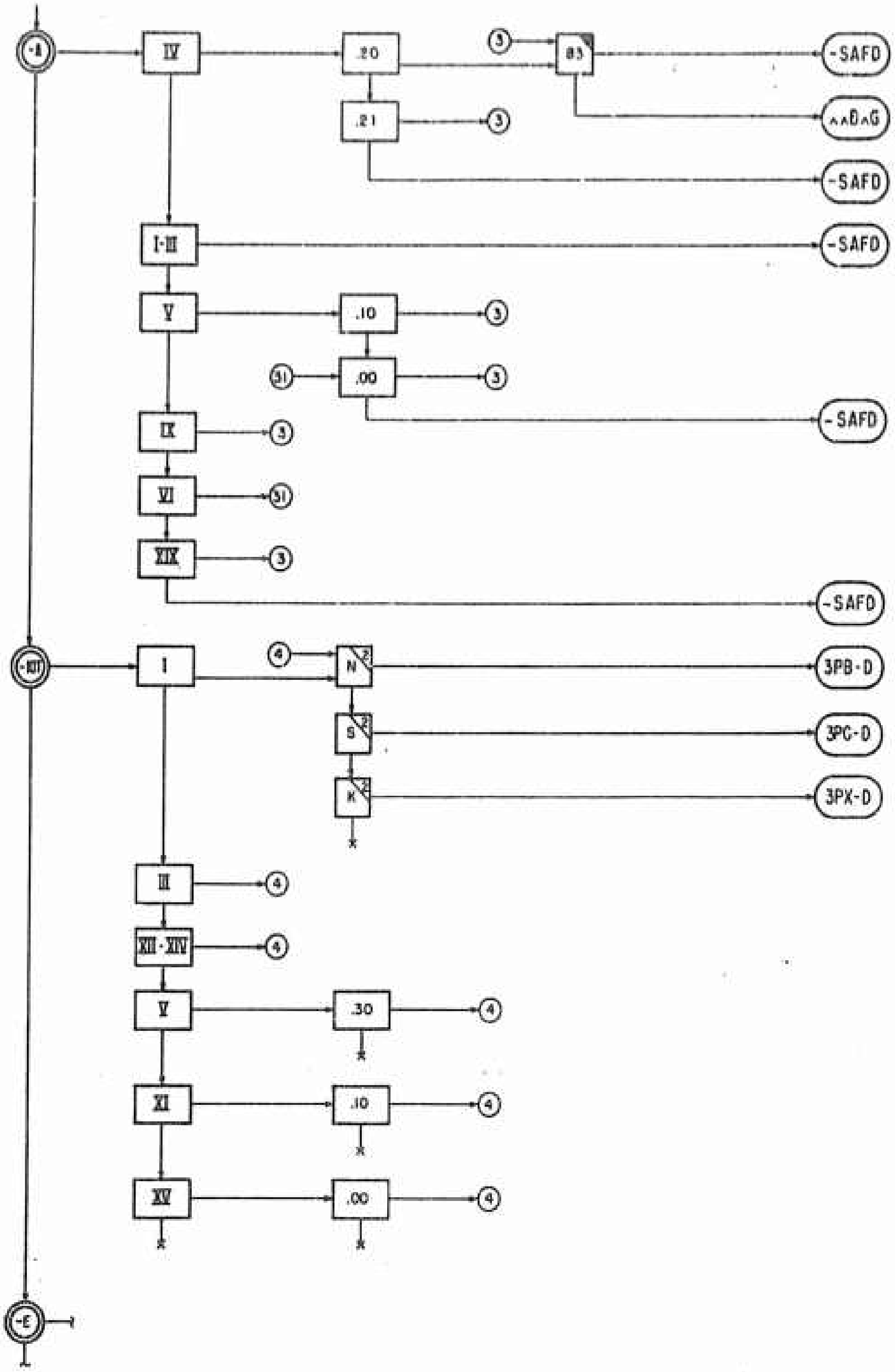
Note 6: Insert "Ns and As" only if affix of canonical form is oй.

Note 7: Mark "INCOMPAT EE" in word 24.

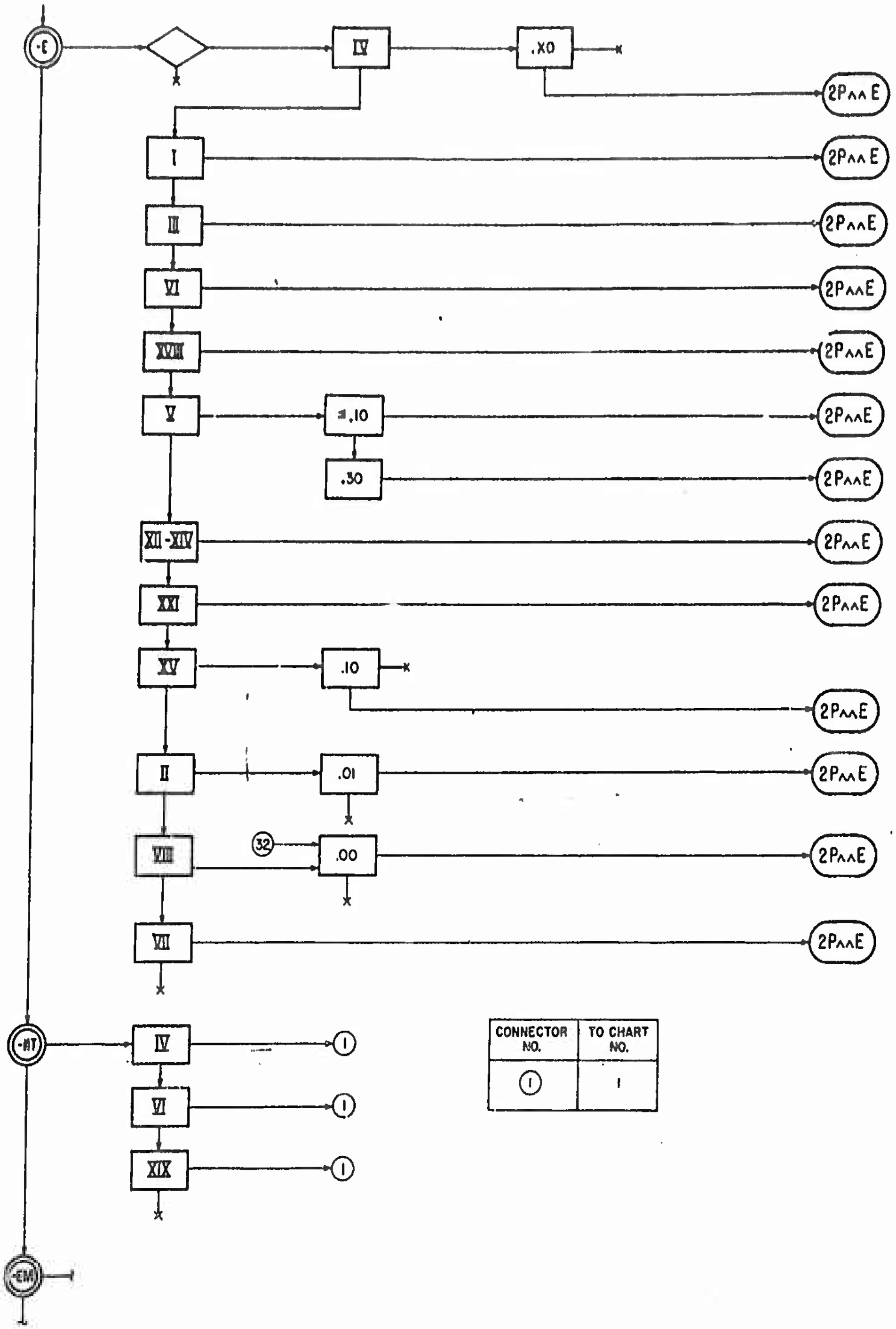


Verb Analyzer Program
Chart No. 1

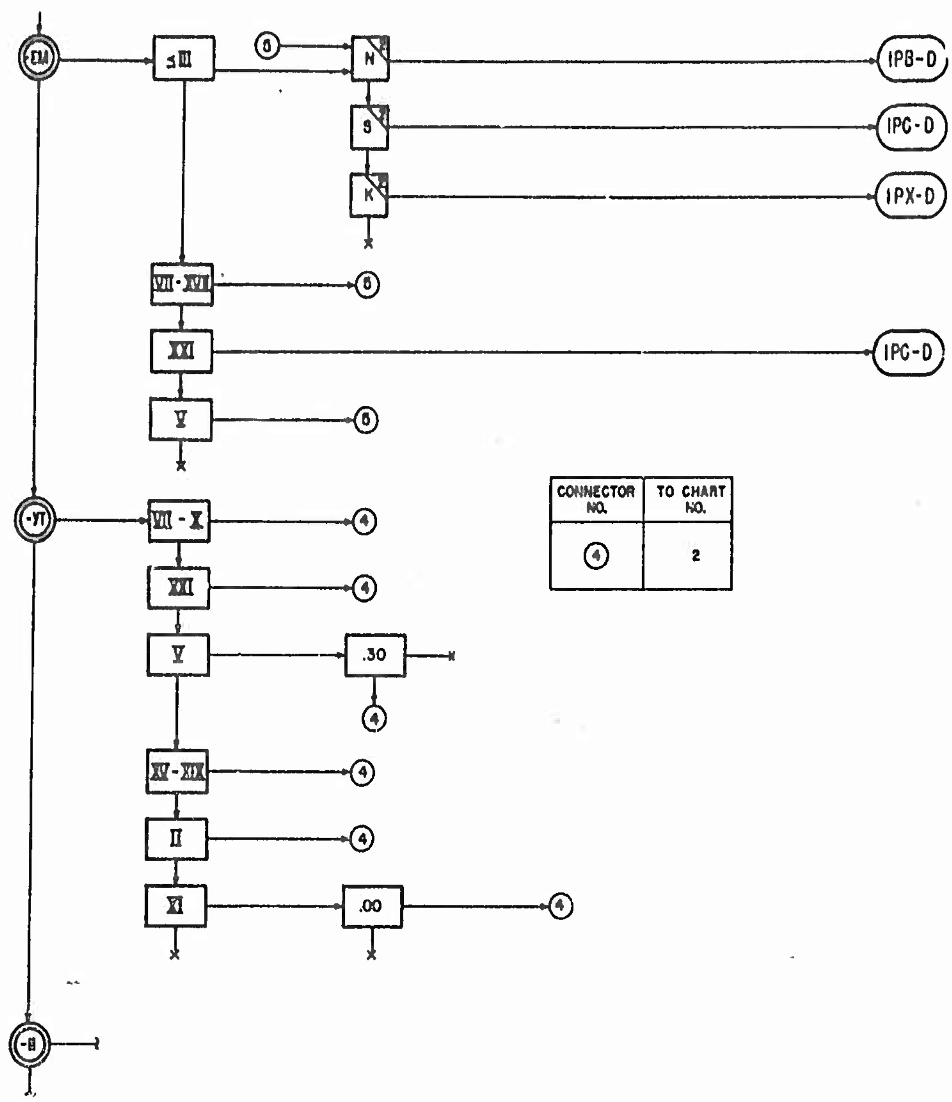
Flow Chart D-3



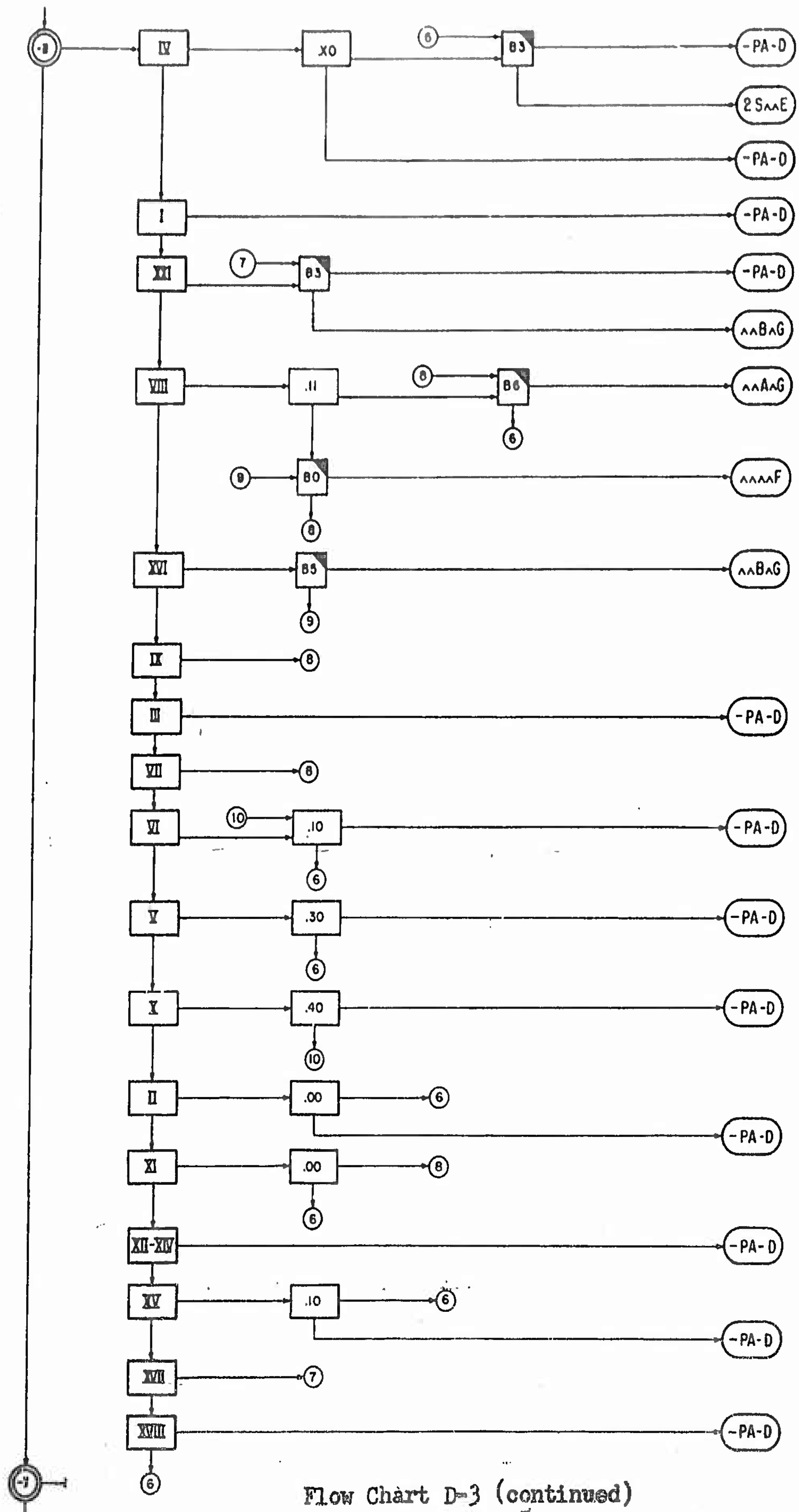
Flow Chart D-3 (continued)
Chart No. 2



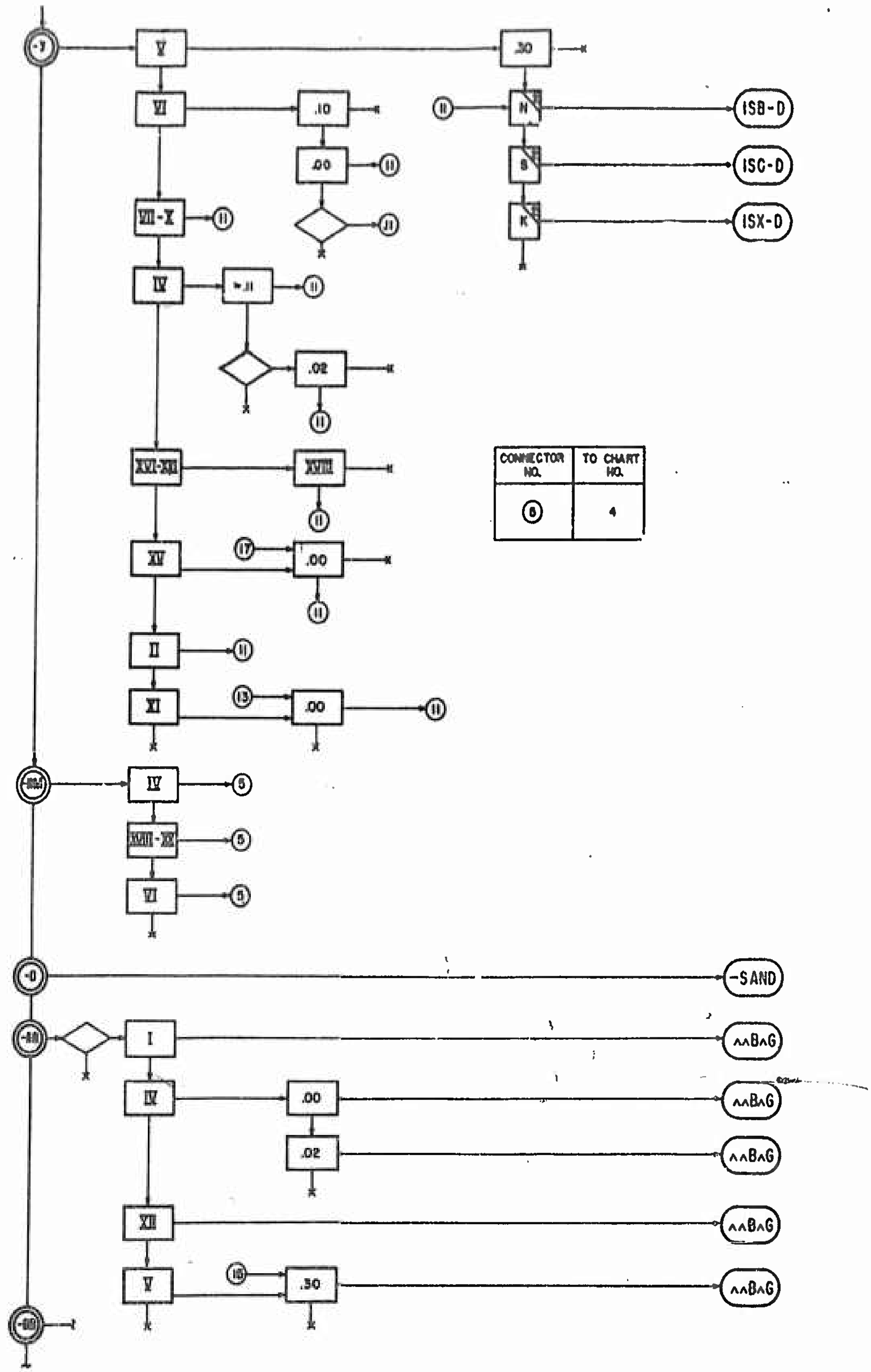
Flow Chart D-3 (continued)
Chart No. 3



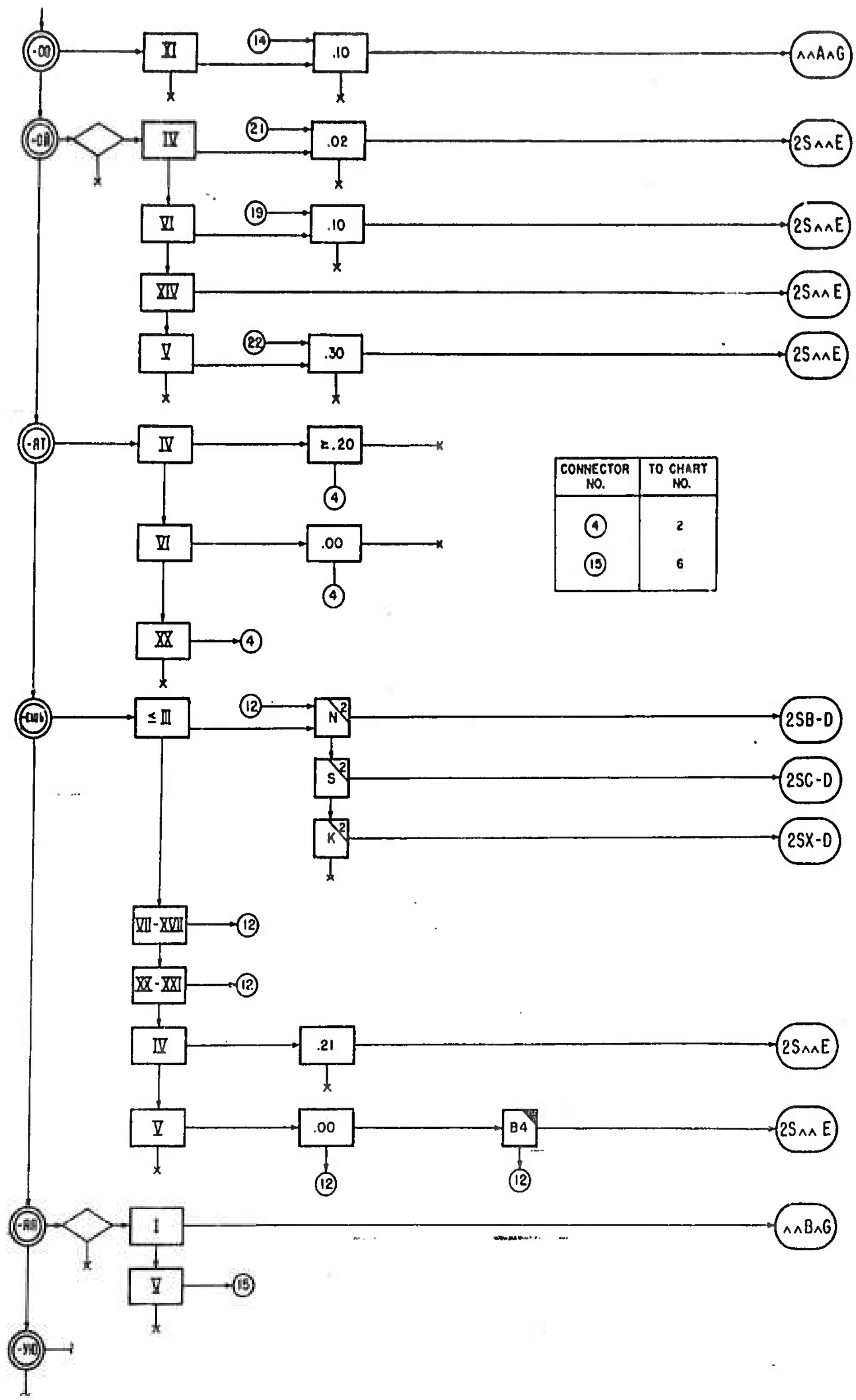
Flow Chart D-3 (continued)
Chart No. 4



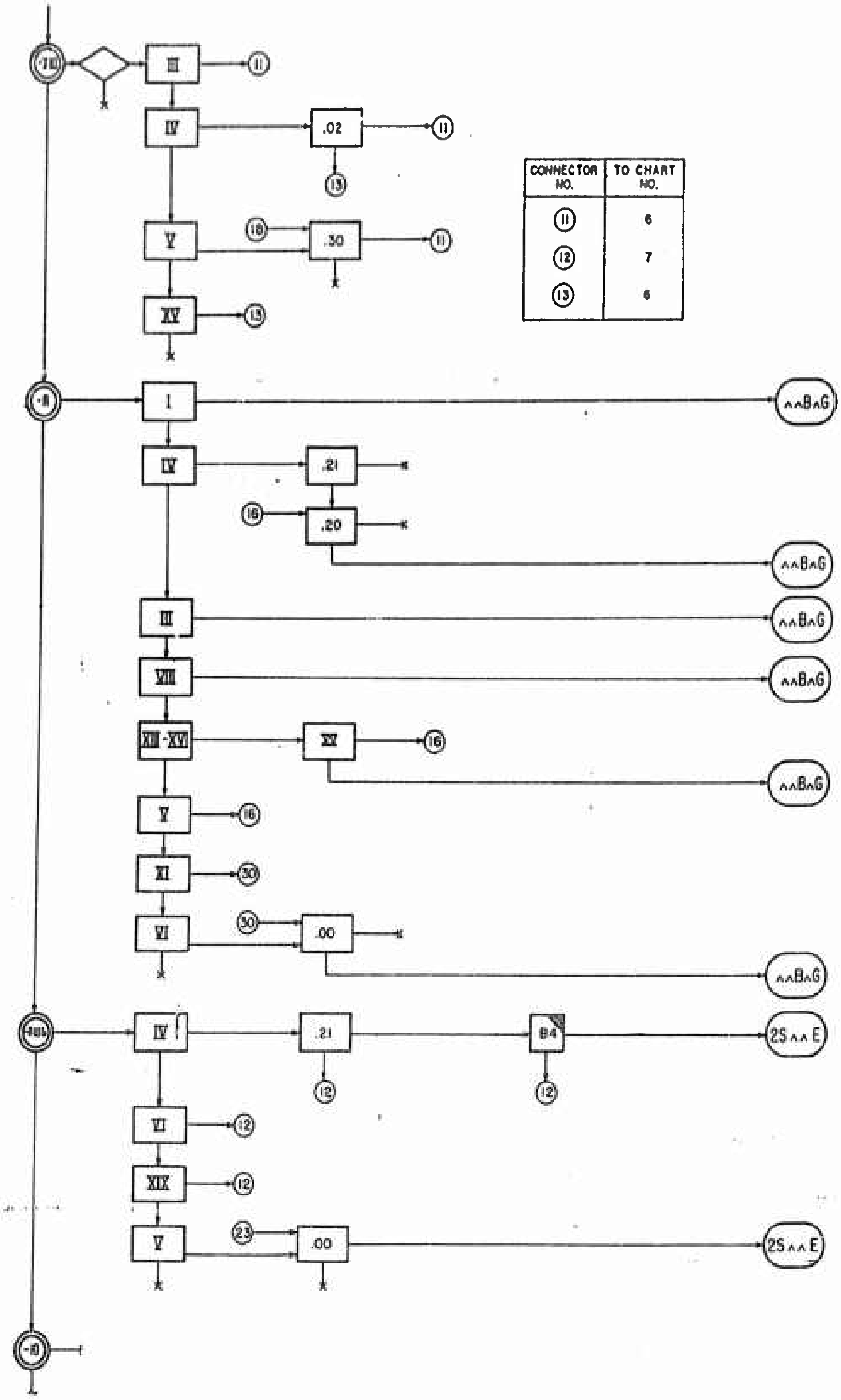
Flow Chart D-3 (continued)
Chart No. 5



Flow Chart D-3 (continued)
Chart No. 6

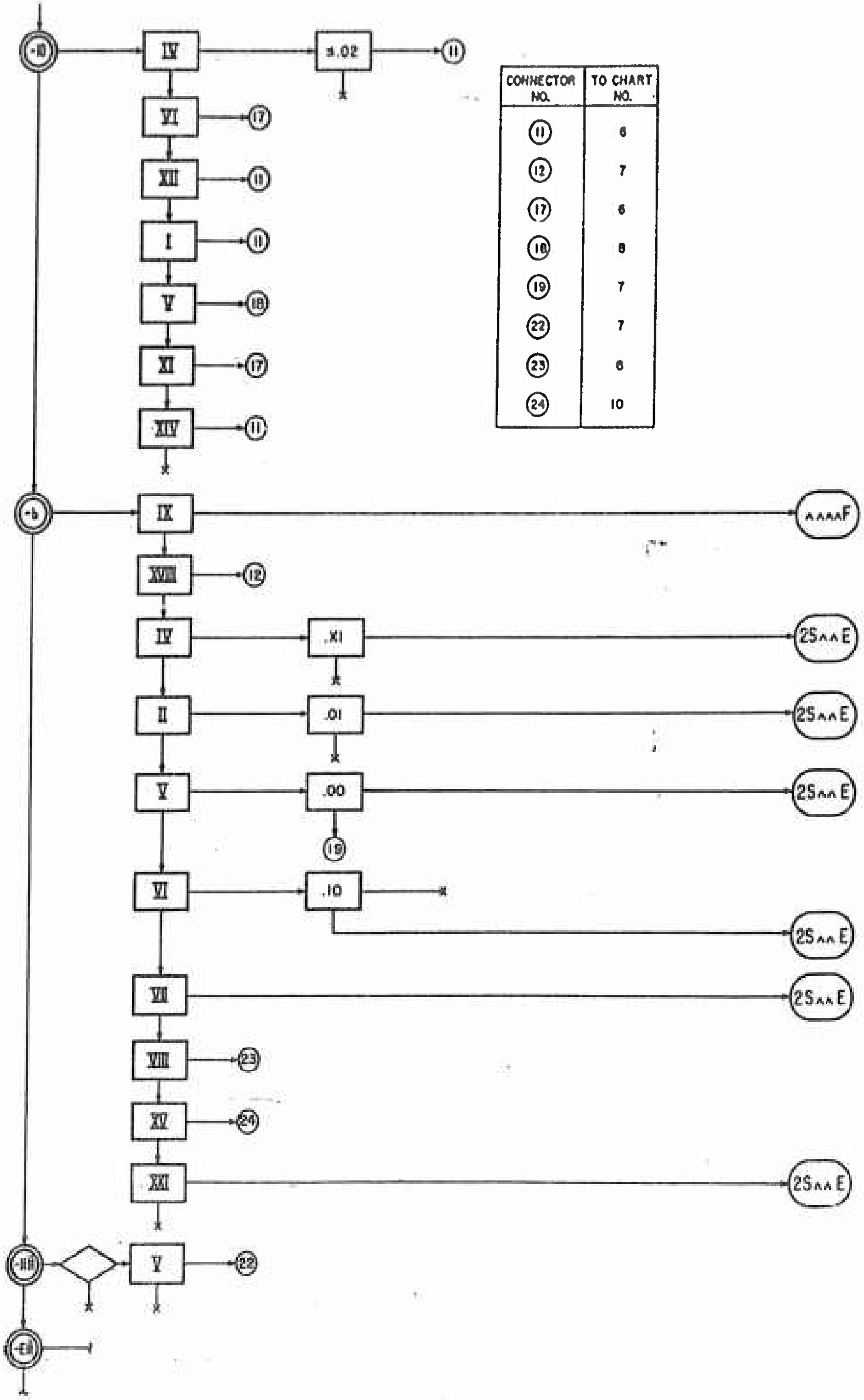


Flow Chart D-3 (continued)
Chart No. 7



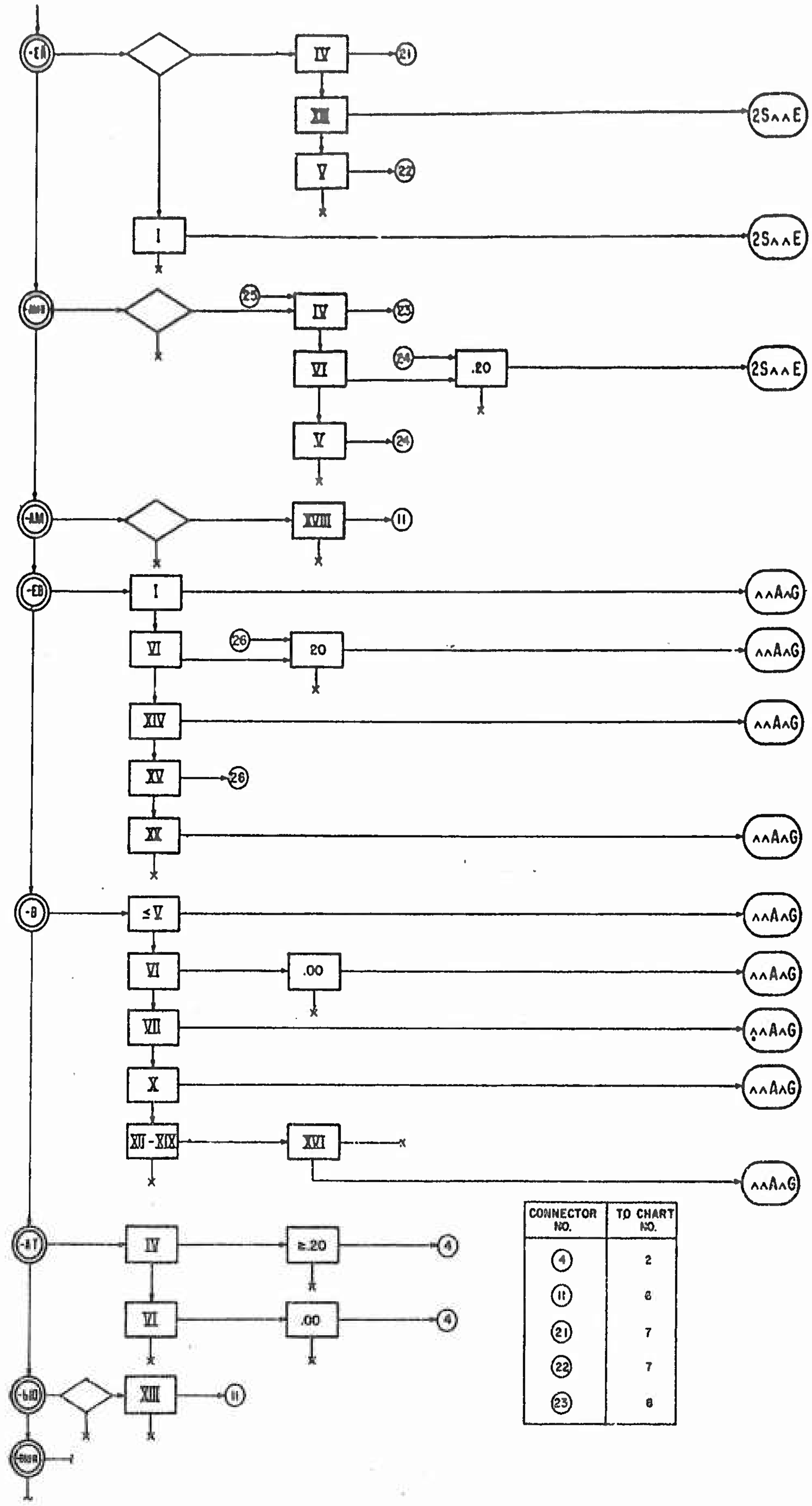
CONNECTOR NO.	TO CHART NO.
11	6
12	7
13	6

Flow Chart D-3 (continued)
Chart No. 8



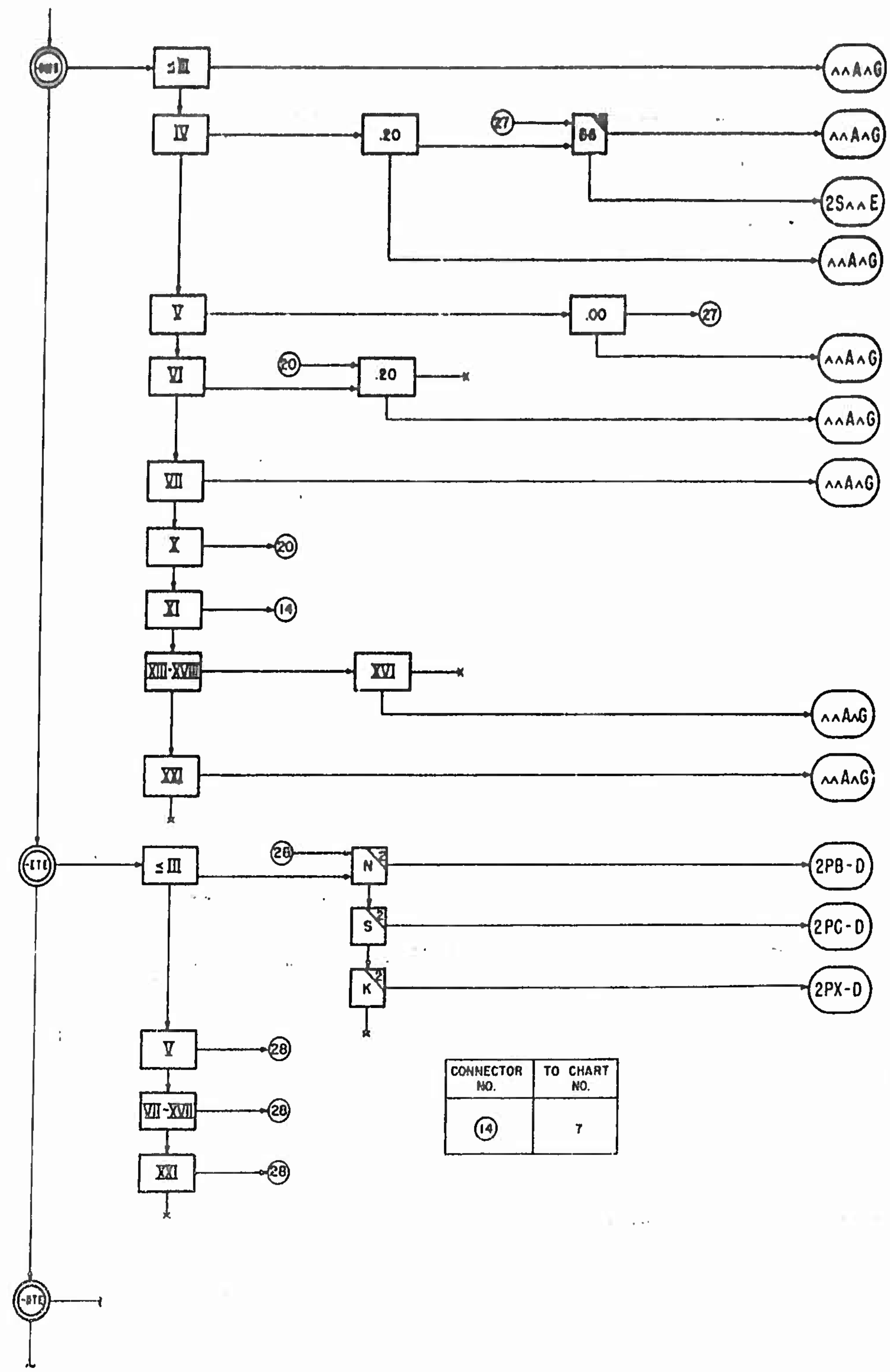
CONNECTOR NO.	TO CHART NO.
11	6
12	7
17	6
18	8
19	7
22	7
23	6
24	10

Flow Chart D-3 (continued)
Chart No. 9



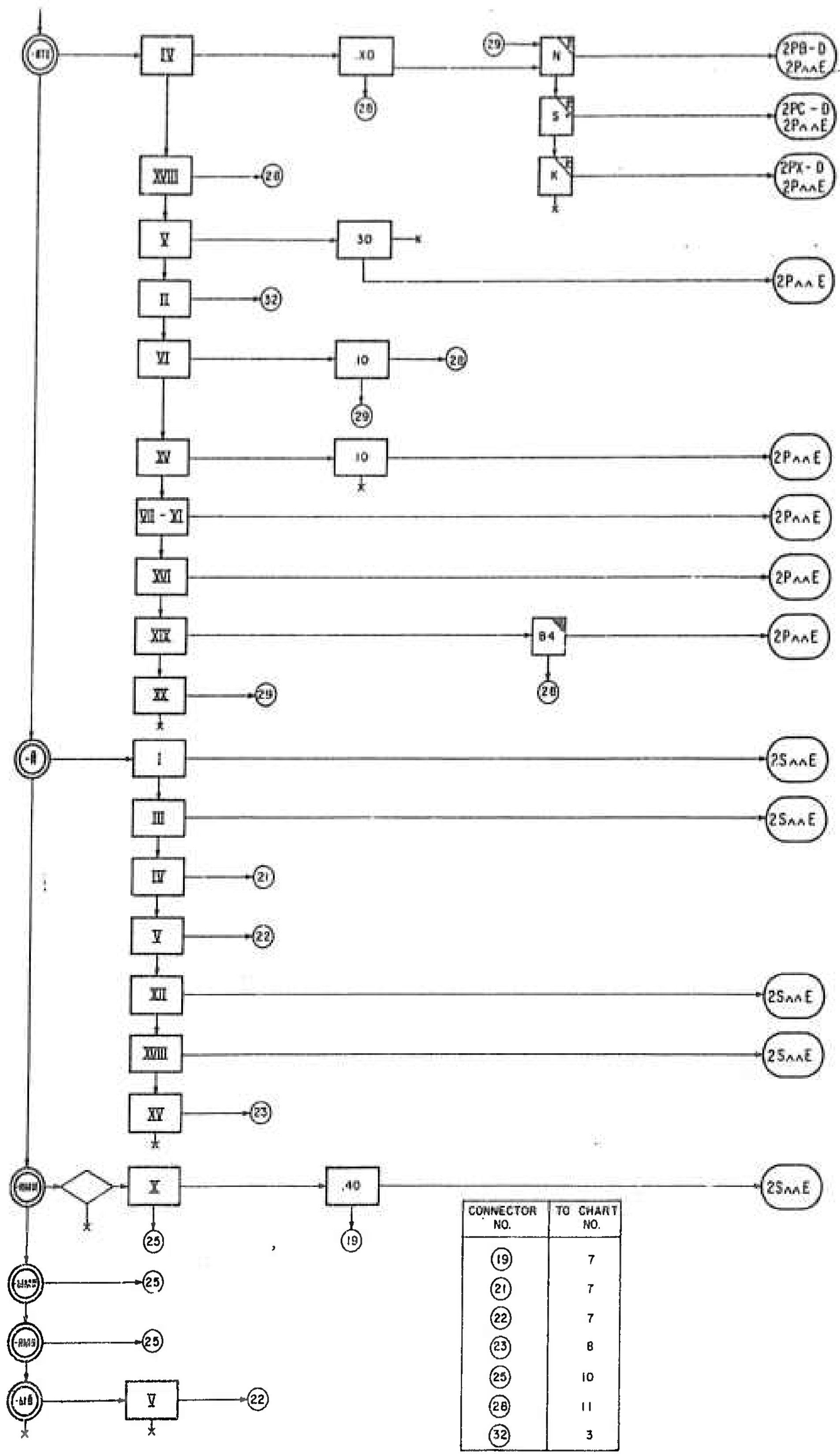
CONNECTOR NO.	TO CHART NO.
4	2
11	6
21	7
22	7
23	6

Flow Chart D-3 (continued)
Chart No. 10



CONNECTOR NO.	TO CHART NO.
14	7

Flow Chart D-3 (continued)
Chart No. 11



CONNECTOR NO.	TO CHART NO.
19	7
21	7
22	7
23	8
25	10
28	11
32	3

Flow Chart D-3 (continued)
Chart No. 12

Appendix E

FREQUENCY OF REFERENCE TO DICTIONARY ENTRIES

AFFIX	CLASS MARKER	PER CLASS MARKER TOTAL	PER CLASS MARKER COMPATIBLE	PER CLASS MARKER INCOMPATIBLE	PER AFFIX TOTAL	PER AFFIX COMPATIBLE	PER AFFIX INCOMPATIBLE
*	01.00	193	193				
	02.00	96	96				
	03.00	361	6	355			
	06.00	2	2				
	08.00	38	38		690	335	355
A	01.00	266	266				
	02.00	163	163				
	03.00	431	40	391			
	06.00	5	5				
	08.00	11	11		876	485	391
AM	03.00	6		6	6		6
AMI	03.00	36		36	36		36
AT	03.00	11		11	11		11
AT SJA	03.00	12		12	12		12
AX	03.00	25		25	25		25
AJA	01.00	187	187				
	01.FT	3	3				
	02.00	226	226				
	03.00	160	152	8			
	04.00	157	136	21			
	05.00	1		1			
	06.00	59	59				
	08.00	17	17		810	780	30
AJA SJA	04.00	27	17	10	27	17	10
V	03.00	4	4		4	4	
E	01.00	4		4			
	03.00	368		368			
	04.00	64	64				
	05.00	6	6				
	08.00	39		39			
	80.00	5	5		486	75	411
EV	03.00	1		1	1		1
EGO	01.00	2		2			
	02.00	1		1			
	04.00	206	186	20			
	05.00	61	61				
	08.00	2		2			
	80.00	12	12		284	259	25
EGO SJA	04.00	43	24	19	43	24	19
EE	01.00	3	3				
	02.00	25	25				
	03.00	25	8	17			
	04.00	191	181	10			
	05.00	44	44				
	08.00	7	7		295	268	27
EE SJA	04.00	26	17	9	26	17	9
EJ	02.00	1		1			
	03.00	104		104			
	04.00	295	266	29			
	05.00	59	59				
	08.00	12		12			
	80.00	2	2		473	327	146
EJ SJA	04.00	26	17	9	26	17	9
EM	01.00	1		1			
	03.00	112	99	13			
	04.00	133	129	4			
	05.00	16	16				
	08.00	2		2			
	80.00	2	2		266	246	20
EM SJA	03.00	13	9	4			
	04.00	2	2		15	11	4
EMU	04.00	23	21	2			
	05.00	5	5		28	26	2
EMU SJA	04.00	5	4	1	5	4	1
ET	03.00	702		702	702		702
ET SJA	02.00	2		2			
	03.00	349		349	351		351
ETE	03.00	3		3	3		3
I	01.00	7		7			
	02.00	1		1			
	03.00	193		193			
	04.00	2	2				
	05.00	1	1				

Frequency of Reference to Adjectival Dictionary Entries

TABLE E-1(a)

AFFIX	CLASS MARKER	PER CLASS MARKER TOTAL	PER CLASS MARKER COMPATIBLE	PER CLASS MARKER INCOMPATIBLE	PER AFFIX TOTAL	PER AFFIX COMPATIBLE	PER AFFIX INCOMPATIBLE
	06.00	91	91				
	08.00	1	1				
	80.00	1	1		297	96	201
IE	03.00	5		5			
	04.00	400	356	44			
	05.00	31	31				
	06.00	110	110				
	08.00	39	39		585	536	49
IE SJA	04.00	105	66	39	105	66	39
IJ	01.00	1		1			
	02.00	1		1			
	04.00	243	215	28			
	05.00	18	18				
	06.00	59	59				
	08.00	16	16		338	308	30
IJ SJA	04.00	38	25	13	38	25	13
IM	02.00	2		2			
	03.00	64	56	8			
	04.00	182	172	10			
	05.00	23	23				
	06.00	70	70				
	08.00	52	52				
	80.00	2	2		395	375	20
IM SJA	03.00	2		2			
	04.00	20	14	6	22	14	8
IMI	03.00	1		1			
	04.00	74	71	3			
	05.00	15	15				
	06.00	36	36				
	07.00	1	1				
	08.00	20	20		147	143	4
IMI SJA	04.00	15	11	4	15	11	4
IT	03.00	211	1	210	211	1	210
IT SJA	01.00	1		1			
	03.00	86	1	85	87	1	86
IX	03.00	53		53			
	04.00	562	487	75			
	05.00	47	47				
	06.00	249	249				
	08.00	68	68		979	851	128
IX SJA	04.00	136	87	49	136	87	49
J	03.00	1		1			
	80.00	2	2		3	2	1
O	01.00	541	539	2			
	02.00	1150	1150				
	03.00	556	335	221			
	04.00	9	9				
	05.00	1		1			
	06.00	115	115				
	08.00	23	23		2395	2171	224
O SJA	03.00	1		1	1		1
OV	03.00	159	97	62	159	97	62
OGO	01.00	329	328	1			
	02.00	555	555				
	03.00	367	366	1			
	04.00	13		13			
	06.00	161	161				
	08.00	58	58		1483	1468	15
OE	01.00	144	142	2			
	01.FT	2	2				
	02.00	476	476				
	03.00	177	172	5			
	04.00	18		18			
	06.00	90	90				
	08.00	42	42		949	924	25
OJ	01.00	403	403				
	01.FT	1	1				
	02.00	762	762				
	03.00	507	482	25			
	04.00	27		27			
	06.00	166	166				
	08.00	111	111		1977	1925	52
OM	01.00	124	124				

TABLE E-1(a) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER TOTAL	PER CLASS MARKER COMPATIBLE	PER CLASS MARKER INCOMPATIBLE	PER AFFIX TOTAL	PER AFFIX COMPATIBLE	PER AFFIX INCOMPATIBLE
	01.FT	1	1				
	02.00	180	180				
	03.00	179	135	44			
	04.00	6		6			
	06.00	38	38				
	08.00	34	34		562	512	50
OHU	01.00	50	50				
	02.00	70	70				
	03.00	40	40				
	04.00	1		1			
	06.00	7	7				
	08.00	19	19		187	186	1
T'	03.00	243		243			
	04.00	373		373	616		616
T' SJA	03.00	98		98	98		98
U	01.00	2		2			
	03.00	90		90			
	08.00	8		8	100		100
UT	02.00	1		1			
	03.00	9	8	1	10	8	2
UT SJA	03.00	1		1	1		1
UJU	01.00	125	125				
	02.00	141	141				
	03.00	89	89				
	04.00	112	102	10			
	06.00	23	23				
	08.00	28	28		518	508	10
UJU SJA	04.00	19	11	8	19	11	8
Y	01.00	268	268				
	02.00	165	165				
	03.00	198	83	115	631	516	115
YE	01.00	299	298	1			
	01.FT	3	3				
	02.00	623	623				
	03.00	290	287	3	1215	1211	4
YJ	01.00	133	133				
	02.00	256	256				
	03.00	97	97		486	486	
YM	01.00	141	141				
	01.FT	1	1				
	02.00	420	420				
	03.00	130	128	2			
	05.00	1		1	693	690	3
YMI	01.00	103	102	1			
	02.00	221	221				
	03.00	94	94		418	417	1
YX	01.00	533	533				
	01.FT	3	3				
	02.00	1186	1186				
	03.00	498	496	2	2220	2218	2
'	03.00	36		36	36		36
'JU	03.00	41		41	41		41
JU	01.00	1		1			
	03.00	29		29	30		30
JUT	03.00	218		218	218		218
JUT SJA	03.00	176		176	176		176
JUJU	01.00	2		2			
	05.00	9	9		11	9	2
JA	03.00	56		56			
	80.00	4	4		60	4	56
JAM	03.00	6		6	6		6
JAMI	03.00	17		17	17		17
JAT	03.00	37	4	33	37	4	33
JAT SJA	03.00	32		32	32		32
JAX	02.00	1		1			
	03.00	9		9	10		10
JAJA	03.00	10		10			
	05.00	31	31		41	31	10
					24312	18807	5505

TABLE E-1(a) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
#	01.00	246	246		246	246	
A	01.00	172	172		172	172	
AX	01.00	1	1		1	1	
E	01.00	79	79		79	79	
EM	01.00	5	5		5	5	
I	01.00	72	72		72	72	
IM	01.00	55	55		55	55	
IMI	01.00	21	21		21	21	
IX	01.00	81	81		81	81	
O	01.00	192	192		192	192	
OV	01.00	3	3		3	3	
OGO	01.00	90	87	3	90	87	3
OE	01.00	4		4	4		4
OJ	01.00	138	106	32	138	106	32
OM	01.00	43	35	8	43	35	8
OMU	01.00	24	24		24	24	
T'	01.00	14	14		14	14	
U	01.00	40	37	3	40	37	3
UJU	01.00	4		4	4		4
Y	01.00	7	5	2	7	5	2
YE	01.00	11		11	11		11
YM	01.00	10		10	10		10
YMI	01.00	6		6	6		6
YX	01.00	22		22	22		22
'	01.00	2	2		2	2	
'JU	01.00	4	4		4	4	
JU	01.00	1	1		1	1	
JA	01.00	34	34		34	34	
					1381	1276	105

Frequency of Reference to Numeric Dictionary Entries

TABLE E-1(b)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
#	01.00	7612	5910	1702	7612	5910	1702
A	00.00	2		2			
	01.00	2490	2352	138	2492	2352	140
AM	01.00	10	10		10	10	
AMI	01.00	28	7	21	28	7	21
AX	01.00	11		11	11		11
AJA	01.00	46	18	28	46	18	28
V	01.00	3763	3761	2	3763	3761	2
E	01.00	2723	2440	283	2723	2440	283
E S'	01.00	86	84	2	86	84	2
EGO	01.00	7	6	1	7	6	1
EE	01.00	252	240	12	252	240	12
EJ	01.00	30		30	30		30
EM	01.00	518	284	234	518	284	234
EMU	01.00	2	2		2	2	
ET	01.00	238	229	9	238	229	9
ET SJA	01.00	1		1	1		1
I	01.00	6604	5114	1490	6604	5114	1490
IE	01.00	4	2	2	4	2	2
IM	01.00	19	13	6	19	13	6
IMI	01.00	2		2	2		2
IT	01.00	24	22	2	24	22	2
IT SJA	01.00	2		2	2		2
IX	01.00	4		4	4		4
ISH'	01.00	127	127		127	127	
O	00.00	5	5				
	01.00	4388	4144	244	4393	4149	244
O SJA	01.00	2		2	2		2
OV	01.00	2		2	2		2
OGO	00.00	12		12			
	01.00	20	2	18	32	2	30
OE	00.00	8		8			
	01.00	32	1	31	40	1	39
OJ	00.00	6		6			
	01.00	54		54	60		60
OM	00.00	1		1			
	01.00	94	85	9	95	85	10
OMU	01.00	158	146	12	158	146	12
T'	01.00	286	230	56	286	230	56
U	01.00	318	283	35	318	283	35
UT	01.00	101	2	99	101	2	99
UJU	00.00	3		3			
	01.00	18	2	16	21	2	19
Y	01.00	432	269	163	432	269	163
YE	01.00	34	9	25	34	9	25
YJ	00.00	3		3			
	01.00	13		13	16		16
YM	00.00	7		7			
	01.00	33		33	40		40
YMI	00.00	1		1			
	01.00	16		16	17		17
YX	01.00	21	2	19	21	2	19
'	01.00	144	137	7	144	137	7
'JU	01.00	29	29		29	29	
JU	01.00	1		1	1		1
JUJU	01.00	1		1	1		1
JA	01.00	1307	1304	3	1307	1304	3
JAMI	01.00	8		8	8		8
JAX	01.00	3		3	3		3
					32166	27271	4895

Frequency of Reference to Indeclinable Dictionary Entries

TABLE E-1(c)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
*	01.00	1750	1724	26			
	01.10	201	201				
	01.20	34	34				
	01.30	48	48				
	02.00	4		4			
	03.05	13		13			
	03.10	7	7				
	04.00	614	613	1			
	04.05	1	1				
	04.10	131	131				
	04.30	77	77				
	04.31	45	45				
	05.05	2		2			
	06.00	12		12			
	08.00	21	21				
	08.15	144	144				
	12.00	2	2		3106	3048	58
A	01.00	2128	2088	40			
	01.10	395	395				
	01.20	34	34				
	01.30	77	77				
	02.00	60		60			
	03.05	12		12			
	03.10	8	8				
	04.00	870	869	1			
	04.05	2	2				
	04.10	156	156				
	04.30	35	35				
	04.31	84	84				
	06.00	1		1			
	07.00	3		3			
	08.00	451	451				
	08.15	222	222				
	10.00	3		3			
	99.90	31	31		4572	4452	120
AM	01.00	69	67	2			
	01.10	9	9				
	01.20	4	4				
	02.00	2		2			
	04.00	77	77				
	04.10	11	11				
	04.30	3	3				
	04.31	3	3				
	08.00	18	18				
	08.15	16	16		212	208	4
AMI	01.00	208	205	3			
	01.10	25	25				
	01.20	16	16				
	01.30	12	12				
	03.10	3	3				
	04.00	116	116				
	04.10	12	12				
	04.30	8	8				
	04.31	11	11				
	07.00	1					
	08.00	64	64				
	08.15	19	19				
	12.00	1	1		496	492	4
AT	01.00	48	48				
	04.00	33	33		81	81	
AX	01.00	148	145	3			
	01.10	14	14				
	01.20	6	6				
	01.30	3	3				
	01.40	1	1				
	04.00	139	139				
	04.10	7	7				
	04.30	50	50				
	04.31	7	7				
	06.00	2		2			
	06.10	1	1				
	08.00	24	24				
	08.15	3	3				

Frequency of Reference to Nominal Dictionary Entries

TABLE E-1(d)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
	10.00	1		1	406	400	6
AJA	01.00	1		1			
	01.10	8		8			
	02.00	40	36	4			
	04.31	1		1			
	08.00	6		6	56	36	20
V	01.00	20	20				
	04.00	15	15				
	08.00	107	107				
	10.00	1		1	143	142	1
VSHIS:	10.00	5		5	5		5
E	01.00	688	593	95			
	01.10	43	43				
	01.20	13	13				
	01.30	49	49				
	02.00	271	271				
	03.00	8	8				
	03.05	8	8				
	03.10	3	3				
	04.00	690	690				
	04.05	2	2				
	04.10	71	71				
	04.30	76	76				
	04.31	55	55				
	05.10	1	1				
	05.20	10	10				
	06.00	1		1			
	08.00	167	167				
	08.15	29	29				
	10.00	1887	1887				
	11.10	1	1				
	13.00	30	30		4103	4007	96
E S'	06.00	2	2		2	2	
EV	01.00	5	5				
	02.00	24	24				
	99.99	10	10		39	39	
EGO.	01.00	6		6			
	03.05	1		1			
	08.00	1		1	8		8
EJ	01.00	9		9			
	03.00	98	98				
	03.05	7	7				
	03.10	5	5				
	04.00	19	19				
	04.10	9	9				
	05.00	8	8				
	05.05	3	2	1			
	05.10	3	3				
	05.20	8		8			
	06.00	375	375				
	07.00	88	88				
	08.00	1		1			
	13.00	3	3		636	617	19
EM	01.00	49	15	34			
	02.00	9	9				
	03.00	10	10				
	04.00	422	211	211			
	06.00	11		11			
	10.00	319	319				
	12.00	4	4				
	13.00	6	6		830	574	256
EMU	01.00	8	4	4			
	04.00	230	115	115			
	07.00	7		7	245	119	126
ET	01.00	129	63	66			
	04.00	6	3	3			
	05.05	22		22			
	06.00	6		6			
	08.00	4	4		167	70	97
ET SJA	01.00	1		1	1		1
ETE	01.00	23	11	12			
	04.00	2	1	1			
	13.00	1		1	26	12	14

TABLE E-1(d) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER TOTAL	PER CLASS MARKER COMPATIBLE	PER CLASS MARKER INCOMPATIBLE	PER AFFIX TOTAL	PER AFFIX COMPATIBLE	PER AFFIX INCOMPATIBLE
I	01.00	876		876			
	01.10	104	104				
	01.30	13	13				
	02.00	28	24	4			
	03.00	47	47				
	03.05	1	1				
	03.10	7	7				
	04.00	47		47			
	04.10	410	410				
	04.30	166	166				
	04.31	165	165				
	05.00	4	4				
	05.05	8	8				
	05.10	15	15				
	05.20	18	18				
	06.00	1630	1630				
	06.10	34	34				
	07.00	1021	1020	1			
	10.00	675	675				
	12.00	114	114		5383	4455	928
I S'	06.00	16	16		16	16	
IE	01.10	27		27			
	04.31	1		1			
	10.00	76	38	38	104	38	66
IJ	02.00	11	6	5			
	04.00	14		14			
	07.00	32	32				
	10.00	9	4	5	66	42	24
IM	01.00	46	28	18			
	01.10	26		26			
	04.00	4		4			
	04.31	1		1			
	07.00	1		1			
	08.00	1		1	79	28	51
IM SJA	01.00	1		1	1		1
IMI	01.10	13		13	13		13
IT	01.00	87	2	85	87	2	85
IT SJA	01.00	32		32			
	02.00	1		1			
	04.00	3		3			
	08.00	7		7	43		43
IX	01.10	48		48	48		48
J	02.00	36	36				
	05.05	6	6				
	07.00	236	236				
	10.00	671	671		949	949	
O	01.00	11		11			
	01.10	2		2			
	04.00	1		1			
	04.10	8		8			
	04.31	3		3			
	06.00	34		34			
	08.00	474	474				
	08.10	1	1				
	08.15	259	259				
	10.00	10		10			
	99.90	6	6		809	740	69
O S'	03.00	1		1	1		1
OV	01.00	1416	1409	7			
	01.10	160	160				
	01.20	46	46				
	01.30	24	24				
	02.00	90		90			
	03.05	5		5			
	04.00	13	3	10			
	04.10	5		5			
	06.00	5		5			
	08.00	45	45		1809	1687	122
OGD	01.10	29		29			
	02.00	8		8			
	04.31	1		1			
	05.05	4		4			
	06.00	6		6	48		48

TABLE E-1(d) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER TOTAL	PER CLASS MARKER COMPATIBLE	PER CLASS MARKER INCOMPATIBLE	PER AFFIX TOTAL	PER AFFIX COMPATIBLE	PER AFFIX INCOMPATIBLE
OE	01.00	10		10			
	01.10	13		13			
	02.00	32	8	24			
	04.31	2		2			
	06.00	11		11			
	08.00	3		3	71	8	63
OJ	01.00	11		11			
	01.10	64		64			
	01.30	1		1			
	02.00	39	15	24			
	04.00	266	266				
	04.05	1	1				
	04.10	10	10				
	04.30	16	16				
	04.31	43	43				
	06.00	1		1	452	351	101
OM	01.00	741	738	3			
	01.10	93	93				
	01.20	16	16				
	01.30	21	21				
	02.00	6		6			
	03.05	7		7			
	03.10	1	1				
	04.00	18	12	6			
	04.10	1		1			
	06.00	12		12			
	08.00	118	118				
	08.15	55	55				
	99.90	3	3		1092	1057	35
OMU	01.00	22	22				
	01.10	15		15			
	02.00	2		2			
	04.00	1		1	40	22	18
T'	01.00	33		33			
	05.00	95	95				
	10.00	2		2	130	95	35
U	01.00	251	239	12			
	01.10	26	26				
	01.20	9	9				
	01.30	16	16				
	03.05	2		2			
	03.10	2	2				
	04.00	365	365				
	04.10	39	39				
	04.30	41	41				
	04.31	44	44				
	08.00	43	43				
	08.15	27	27				
	99.90	2	2		867	853	14
UT	01.00	10	10				
	05.05	2		2			
	06.00	1		1	13	10	3
UT SJA	01.00	1		1	1		1
UJU	01.10	8		8			
	02.00	2		2			
	06.00	1		1	11		11
UJU S'	04.00	3		3	3		3
Y	01.00	866	848	18			
	01.20	25	25				
	03.05	2		2			
	04.00	1829	1829				
	04.05	5	5				
	06.00	2		2	2729	2707	22
YE	02.00	10		10			
	04.00	2		2			
	06.00	2		2	14		14
YJ	06.00	4		4	4		4
YM	02.00	4		4			
	04.00	2		2			
	06.00	2		2	8		8
YMI	02.00	10		10			
	04.00	2		2	12		12
YX	02.00	10		10			

TABLE E-1(d) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER TOTAL	PER CLASS MARKER COMPATIBLE	PER CLASS MARKER INCOMPATIBLE	PER AFFIX TOTAL	PER AFFIX COMPATIBLE	PER AFFIX INCOMPATIBLE
	04.00	1		1			
	06.00	7		7	18		18
'	01.00	9		9			
	03.00	100	100				
	03.05	8	8				
	03.10	1		1			
	05.00	1	1				
	05.10	2		2			
	06.00	673	673				
	06.10	17	17		811	799	12
'JU	05.20	1		1			
	06.00	189	189				
	06.10	63	63		253	252	1
•••	04.05	1		1	1		1
JU	01.00	9		9			
	02.00	10	10				
	03.00	16	16				
	03.05	3	3				
	05.00	2	2				
	05.05	1	1				
	07.00	183	183				
	10.00	314	314				
	13.00	3	3		541	532	9
JA	01.00	78		78			
	01.20	1		1			
	02.00	29	29				
	03.00	100	100				
	03.05	25	25				
	05.00	2	2				
	05.05	1	1				
	05.10	1	1				
	05.20	9	9				
	06.00	3		3			
	07.00	328	328				
	08.00	1		1			
	10.00	2912	2912				
	11.10	1	1				
	12.00	63	63				
	13.00	18	18				
	99.90	3	3		3575	3492	83
JAH	02.00	1	1				
	03.05	1	1				
	06.00	24	24				
	07.00	10	10				
	10.00	78	78		114	114	
JAHI	02.00	2	2				
	03.00	17	17				
	03.05	3	3				
	05.00	2	2				
	06.00	54	54				
	07.00	45	45				
	10.00	90	90				
	99.90	4	4		217	217	
JAT	01.00	18		18	18		18
JAT SJA	01.00	17		17			
	08.00	4		4	21		21
JÄK	01.00	1		1			
	02.00	53	53				
	03.00	4	4				
	03.05	3	3				
	05.05	2	2				
	05.20	1	1				
	06.00	62	62				
	07.00	24	24				
	10.00	115	115				
	99.90	1	1		266	265	1
JAJA	04.00	3		3	3		3
					35875	33030	2845

TABLE E-1(d) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
#	01.00	369	369		369	369	
A	01.00	145	145		145	145	
AM	01.00	21	21		21	21	
AMI	01.00	36	36		36	36	
AJA	01.00	205	205		205	205	
V SJA	01.00	10	10		10	10	
E	01.00	257	257		257	257	
E S'	01.00	12	12		12	12	
EGO	01.00	462	462		462	462	
EE	01.00	414	414		414	414	
EJ	01.00	85	85		85	85	
EM	01.00	186	186		186	186	
FMU	01.00	35	35		35	35	
I	01.00	196	196		196	196	
IE	01.00	70	70		70	70	
IJ	01.00	7	7		7	7	
IM	01.00	234	234		234	234	
IMI	01.00	49	49		49	49	
IX	01.00	781	781				
	01.00	81	81		862	862	
O	01.00	1844	1844		1844	1844	
OV	01.00	2	2		2	2	
OGO	01.00	471	471		471	471	
OE	01.00	171	171		171	171	
OJ	01.00	586	585	1	586	585	1
OM	01.00	447	447		447	447	
OMU	01.00	74	74		74	74	
U	01.00	35	35		35	35	
UJU	01.00	48	48		48	48	
Y	01.00	306	306		306	306	
YE	01.00	180	180		180	180	
YJ	01.00	107	107		107	107	
YM	01.00	36	36		36	36	
YMI	01.00	24	24		24	24	
YX	01.00	203	202	1	203	202	1
'	01.00	2	2		2	2	
JU	01.00	27	27		27	27	
JA	01.00	7	7		7	7	
					8225	8223	2

Frequency of Reference to Pronominal Dictionary Entries

TABLE E-1(e)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
Ø	01.00	60	13	47			
	02.00	1		1			
	03.00	91	8	83			
	04.00	125	14	111			
	04.01	119	4	115			
	04.02	2		2			
	04.10	2	1	1			
	04.20	15	8	7			
	05.00	11	6	5			
	06.10	2	2				
	06.20	111		111			
	07.00	1	1				
	09.00	1	1				
	09.10	2	2				
	10.00	2	1	1			
	10.10	1	1				
	10.40	7		7			
	13.00	2		2			
	16.00	6	5	1			
	18.00	129	6	123			
	21.00	38	38		728	111	617
SJA	01.00	18	12	6			
	03.00	7	6	1			
	04.00	1	1				
	04.01	1	1				
	04.21	2	1	1			
	05.00	1	1				
	12.00	1	1				
	14.00	4	2	2			
	18.00	1	1		36	26	10
A	01.00	192	10	182			
	02.00	2		2			
	03.00	65	2	63			
	04.00	360	18	342			
	04.01	138		138			
	04.02	2	1	1			
	04.10	2		2			
	04.20	10	7	3			
	04.21	1	1				
	05.00	2	1	1			
	05.40	2		2			
	05.41	4		4			
	06.00	2	2				
	06.10	2	2				
	06.20	41	1	40			
	07.00	3	3				
	08.20	2	2				
	09.00	5	5				
	10.00	12	6	6			
	14.00	6	3	3			
	16.00	1	1				
	18.00	2	1	1			
	19.00	1		1			
	21.00	47	47		904	113	791
A S'	01.00	18	12	6			
	03.00	1	1				
	04.00	5	5				
	04.01	2	1	1			
	04.20	4	2	2			
	05.00	7	4	3			
	06.00	1	1				
	08.20	1	1				
	15.20	1	1		40	28	12
AM	03.00	2		2			
	04.00	8		8	10		10
AMI	01.00	3		3			
	03.00	3		3			
	04.00	27		27			
	04.01	7		7			
	04.20	3		3			
	10.00	4		4	47		47
AT	04.20	6	6				
	04.21	2	2				

Frequency of Reference to Verbal Dictionary Entries

TABLE E-1(f)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
	06.00	20	20		28	28	
AT SJA	01.00	4		4			
	04.20	4	2	2			
	06.00	12	12		20	14	6
AX	01.00	6		6			
	03.00	5		5			
	04.00	11		11			
	04.01	6		6			
	04.10	1		1	29		29
AJA	01.00	190	162	28			
	04.00	25		25			
	04.01	3		3			
	04.20	11		11			
	04.21	17		17			
	08.20	1		1			
	12.00	2	1	1	249	163	86
AJA S'	01.00	21	15	6			
	04.00	2		2			
	04.01	1		1			
	04.10	4		4			
	04.20	4		4			
	12.00	5	3	2	37	18	19
V	01.00	1	1				
	02.00	6	4	2			
	04.00	28	20	8			
	04.01	7	7				
	04.20	1	1				
	04.21	4	2	2			
	10.00	2	1	1			
	10.30	6	3	3			
	10.40	2	2				
	99.90	1	1		58	42	16
VSHIS'	03.00	3	3				
	04.01	5	5		8	8	
E	01.00	368		368			
	03.00	33		33			
	04.00	126	15	111			
	04.01	189		189			
	06.20	111		111			
	10.00	12		12	839	15	824
E S'	04.01	12	12		12	12	
EV	01.00	28	28		28	28	
EE	02.00	4		4			
	03.00	1		1			
	04.00	1		1			
	07.00	1		1			
	18.00	82		82	89		89
EJ	03.00	1		1			
	04.00	3		3			
	04.20	5		5			
	08.20	4		4			
	16.00	4		4	17		17
EM	01.00	291	182	109			
	02.00	15	10	5			
	03.00	42	35	7			
	04.00	188		188			
	04.01	16		16			
	04.20	1		1			
	05.00	105	64	41			
	05.30	39		39			
	05.40	34	17	17			
	05.41	33	33				
	06.10	2		2			
	08.00	3	1	2			
	08.20	73	67	6			
	09.00	39	39				
	10.00	8	4	4			
	10.30	22	11	11			
	10.40	10	10				
	14.00	1	1				
	16.00	21	19	2			
	18.00	2		2			
	21.00	188	188				
	99.90	4	4		1137	685	452

TABLE E-1(f) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
EH SJA	01.00	23	13	10			
	03.00	9	9				
	05.00	1	1		33	23	10
ET	01.00	1241	929	312			
	02.00	8	5	3			
	03.00	323	309	14			
	04.00	198		198			
	04.20	10		10			
	05.00	8	5	3			
	05.40	2	1	1			
	05.41	1	1				
	07.00	1	1				
	08.00	1	1				
	08.20	20	20				
	09.00	315	315				
	09.10	5	5				
	10.40	6	6				
	12.00	83	78	5			
	14.00	2	1	1			
	15.20	12	5	7			
	16.00	20	19	1			
	18.00	123		123			
	21.00	194	194				
	99.90	4	4		2577	1899	678
ET SJA	01.00	1463	1062	401			
	02.00	2	1	1			
	03.00	101	94	7			
	05.00	33	18	15			
	05.10	3	3				
	05.20	1	1				
	05.40	10	5	5			
	08.20	2	2				
	12.00	68	52	16			
	15.20	20		20			
	16.00	18	11	7			
	18.00	55		55	1776	1249	527
ETE	04.20	3		3	3		3
I	01.00	55	18	37			
	03.00	11		11			
	04.00	25	20	5			
	04.01	23	4	19			
	04.02	2	1	1			
	04.20	20	14	6			
	04.21	1		1			
	05.00	13	7	6			
	05.40	2	1	1			
	05.41	1	1				
	06.10	2	2				
	06.20	3	3				
	08.00	7	7				
	08.20	72	52	20			
	09.00	5	5				
	09.10	1	1				
	10.30	2	1	1			
	10.40	863		863			
	14.00	2		2			
	15.20	3	3				
	16.00	51	36	15			
	18.00	2	1	1			
	21.00	45	45		1211	222	989
I S'	01.00	34	25	9			
	03.00	1	1				
	04.00	6	4	2			
	04.10	2	1	1			
	05.00	2	1	1			
	05.40	4	2	2			
	06.00	2	2				
	08.20	1	1				
	15.20	2	2				
	16.00	2	1	1			
	18.00	2	1	1	58	41	17
IE	01.00	32		32			
	04.20	14		14			
	04.21	5		5			
	16.00	3		3	54		54
IJ	03.00	17		17			
	04.01	2		2			
	07.00	1		1	20		20
IM	01.00	286		286			
	04.00	144	105	39			
	04.01	79	73	6			
	04.02	23	13	10			

TABLE E-1(f) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER TOTAL	COMPATIBLE	INCOMPATIBLE	PER AFFIX TOTAL	COMPATIBLE	INCOMPATIBLE
	04.10	11	11				
	04.20	213	127	86			
	04.21	95	95				
	05.00	22		22			
	06.20	97	97				
	12.00	3		3			
	18.00	9	4	5			
	20.00	1	1		983	526	457
IM SJA	04.01	20	18	2			
	04.10	3	3				
	04.21	6	3	3			
	18.00	6	3	3	35	27	8
IMI	08.00	1		1	1		1
IT	01.00	13		13			
	04.00	198	190	8			
	04.01	14	14				
	04.02	15	15				
	04.20	22	20	2			
	04.21	63	40	23			
	06.00	108	108				
	06.10	148	148				
	06.20	61	61				
	99.99	1	1		643	597	46
IT SJA	01.00	45		45			
	04.00	297	153	144			
	04.01	13	11	2			
	04.02	4	4				
	04.10	4	3	1			
	04.20	34	18	16			
	04.21	8	4	4			
	06.00	11	11				
	06.10	1	1				
	99.90	1	1		418	206	212
ITE	04.00	1		1	1		1
IX	07.00	1		1	1		1
J	01.00	35	35				
	04.01	69		69	104	35	69
O	01.00	96	12	84			
	02.00	6		6			
	03.00	10	8	2			
	04.00	230	14	216			
	04.01	17	5	12			
	04.20	2	1	1			
	04.21	1	1				
	05.00	2	1	1			
	05.30	5		5			
	05.41	1	1				
	06.00	2	2				
	07.00	1	1				
	08.20	6	6				
	09.00	5	5				
	10.00	18	9	9			
	14.00	2	1	1			
	15.20	6	6				
	16.00	4	4				
	21.00	89	89		503	166	337
O S'	01.00	35	29	6			
	03.00	9	9				
	04.00	11	7				
	05.00	13	7	6			
	08.20	1	1				
	12.00	1	1				
	15.20	3	3				
	18.00	2	2		75	59	16
O SJA	02.01	1		1			
	04.00	4	4				
	10.40	1		1	6	4	2
OV	01.00	2		2			
	03.00	48		48			
	04.00	58		58			
	04.01	60		60			
	04.10	1		1			
	05.41	1		1			
	06.20	20		20			
	15.20	1		1	191		191
OGO	01.00	4		4			
	04.00	3		3			
	04.01	2		2			
	04.02	4		4	13		13
OE	01.00	16		16			
	99.90	5		5	21		21
OJ	01.00	45		45			
	03.00	11		11			

TABLE E-1(f) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
	04.00	59		59			
	04.01	2		2			
	04.10	9		9			
	09.10	2		2			
	13.00	1		1	129		129
OM	01.00	12		12			
	03.00	395		395			
	04.00	54		54			
	04.01	17		17			
	06.20	7		7			
	10.00	8		8	493		493
T'	01.00	854	558	296			
	02.00	9	8	1			
	02.01	1	1				
	03.00	133	128	5			
	04.00	397	293	104			
	04.01	107	88	19			
	04.02	54	33	21			
	04.10	12	9	3			
	04.20	151	97	54			
	04.21	34	21	13			
	05.00	125	79	46			
	05.10	8	8				
	05.40	33	18	15			
	05.41	3	3				
	06.00	21	21				
	06.10	10	10				
	06.20	76	76				
	07.00	1	1				
	08.11	9	9				
	08.20	2	2				
	10.00	6	3	3			
	10.10	1	1				
	10.20	2	2				
	10.30	12	6	6			
	10.40	5	5				
	12.00	43	10	33			
	14.00	2	2				
	15.20	3	3				
	18.00	34	22	12			
	19.00	2	2				
	21.00	480	480				
	99.99	6	6		2636	2005	631
T' SJA	01.00	224	169	55			
	02.00	6	3	3			
	03.00	16	16				
	04.00	33	23	10			
	04.01	8	5	3			
	04.02	2	2				
	04.10	11	11				
	04.20	6	3	3			
	04.21	12	6	6			
	05.00	27	14	13			
	05.40	4	2	2			
	06.00	2	2				
	10.00	2	1	1			
	12.00	12	7	5			
	13.00	4	4				
	15.20	1	1				
	18.00	2	1	1	372	270	102
U	01.00	32		32			
	03.00	23		23			
	04.00	56	1	55			
	04.01	7		7			
	04.20	1	1				
	06.20	27		27			
	10.00	4		4			
	21.00	1	1		151	3	148
UT	01.00	1		1			
	02.00	12	8	4			
	04.00	98		98			
	05.00	10	5	5			
	05.40	2	1	1			
	07.00	1	1				
	08.00	1	1				
	08.20	3	3				
	09.00	151	151				
	10.40	1	1				
	16.00	1	1				
	18.00	1	1				
	21.00	98	98		380	271	109
UT SJA	05.00	18	10	8			

TABLE E-1(f) (continued)

AFFIX	CLASS MARKER	PER CLASS MARKER			PER AFFIX		
		TOTAL	COMPATIBLE	INCOMPATIBLE	TOTAL	COMPATIBLE	INCOMPATIBLE
	05.40	8	4	4			
	08.20	1	1				
	15.20	1	1				
	16.00	6	6		34	22	12
UJU	01.00	6		6			
	04.00	2		2	8		8
UJU S'	03.00	3	3		3	3	
Y	01.00	145		145			
	03.00	38		38			
	04.00	56		56			
	04.01	31		31			
	04.10	1		1			
	06.20	3		3	274		274
YE	01.00	12		12			
	04.00	12		12	24		24
YJ	01.00	6		6	6		6
YM	01.00	32		32			
	04.00	4		4	36		36
YMI	01.00	18		18			
	04.00	3		3	21		21
YX	01.00	14		14			
	04.00	25		25			
	04.01	2		2	41		41
'	01.00	3		3			
	03.00	9		9			
	04.00	7		7			
	04.20	1		1			
	08.20	3		3			
	09.00	4	4				
	09.10	3	3				
	11.10	1		1			
	21.00	2	2				
	99.99	3	3		36	12	24
JU	01.00	14	12	2			
	04.01	30	1	29			
	06.10	1	1		45	14	31
JUT	01.00	415	317	98			
	03.00	125	99	26			
	12.00	24	24				
	15.20	1		1			
	18.00	32		32	597	440	157
JUT SJA	01.00	633	443	190			
	03.00	42	40	2			
	12.00	17	12	5			
	15.20	2		2			
	18.00	14		14	708	495	213
JA	01.00	10	4	6			
	03.00	32	24	8			
	04.00	2546	73	2473			
	04.01	142	2	140			
	04.02	2	1	1			
	04.20	1		1			
	05.40	2	1	1			
	05.10	1	1				
	06.20	1	1				
	08.20	10	9	1			
	14.00	1		1			
	16.00	1	1				
	20.00	32		32	2781	117	2664
JA S'	03.00	18	18		18	18	
JAM	01.00	1		1			
	04.01	20		20	21		21
JAMI	01.00	4		4			
	03.00	1		1			
	04.01	15		15	20		20
JAT	01.00	1		1			
	04.00	46	42	4			
	04.01	1	1				
	04.02	4	4				
	06.10	16	16				
	06.20	17	17				
	08.00	1		1	86	80	6
JAT SJA	01.00	10		10			
	04.00	82	44	38			
	04.01	7	5	2			
	04.02	7	7		106	56	50
JAX	01.00	51		51			
	04.01	26		26	77		77
JAJA	01.00	66	47	19			
	04.00	32		32			
	04.01	19		19	117	47	70
JAJAS'	01.00	2	2		2	2	
					22265	10200	12065

TABLE E-1(f) (continued)

Nouns							
Affix	Frequency	Percent	Cumulative Percentage	Affix	Frequency	Percent	Cumulative Percentage
и	5383	15.0	15.0	ым	79	.2	98.2
а	4572	12.8	27.8	ое	71	.2	98.4
е	4103	11.4	39.2	оя	67	.2	98.6
я	3575	9.9	49.1	ий	66	.2	98.8
#	3106	8.7	57.8	ая	56	.2	99.0
ы	2729	7.6	65.4	их	48		
ов	1809	5.0	70.4	ого	48		
ом	1092	3.0	73.4	ому	40		
й	949	2.6	76.0	ев	39		
у	867	2.4	78.4	оь	27	.6	99.6
ем	830	2.3	80.7	ете	26		
ь	811	2.3	83.0	ых	18		
о	809	2.3	85.3	ят	18		
ей	636	1.8	87.1	ые	14		
ю	541	1.5	88.6	ими	13	.2	99.8
ами	496	1.4	90.0	ут	13		
ой	452	1.3	91.3	ьми	12		
ах	406	1.1	92.4	ую	11		
ях	266	.7	93.1	его	8		
ью	253	.7	93.8	ьм	8		
ему	245	.7	94.5	ый	4		
ями	217	.6	95.1	яя	3		
ам	212	.6	95.7	—	1	.2	100.0
ет	167	.5	96.2		35875		
в	143	.4	96.6				
ть	130	.4	97.0				
ям	114	.3	97.3				
ие	104	.3	97.6				
ит	87	.2	97.8				
ат	81	.2	98.0				

Frequency of Reference to Nominal Dictionary Entries by Affix (both Compatible and Incompatible) in Frequency Run V

TABLE E-2

Adjectives							
Affix	Frequency	Percent	Cumulative Percentage	Affix	Frequency	Percent	Cumulative Percentage
о	2396	9.9	9.9	ими	162	.7	97.0
ых	2220	9.1	19.0	ов	159	.7	97.7
ой	1977	8.1	27.1	у	100	.4	98.1
ого	1483	6.1	33.2	ят	69	.3	98.4
ые	1215	5.0	38.2	я	60	.2	98.6
их	1115	4.6	42.8	яя	41		
ет	1053	4.3	47.1	ью	41		
ое	949	3.9	51.0	ами	36		
а	876	3.6	54.6	ь	36		
ая	837	3.4	58.0	ему	33	.8	99.4
ть	714	3.0	61.0	ю	30		
ым	693	2.9	63.9	ах	25		
#	690	2.8	66.7	ат	23		
ие	690	2.8	69.5	ами	17		
ы	631	2.6	72.1	ут	11	.4	99.8
ом	562	2.3	74.4	юю	11		
ую	537	2.2	76.6	ях	10		
ей	499	2.1	78.7	ам	6		
е	486	2.0	80.7	ям	6		
ый	486	2.0	82.7	в	4		
ыми	418	1.7	84.4	ете	3		
им	417	1.7	86.1	й	3		
ют	394	1.6	87.7	ев	1	.2	100.0
ий	376	1.6	89.3		24312		
его	327	1.3	90.6				
ее	321	1.3	91.9				
ит	298	1.2	93.1				
и	297	1.2	94.3				
ем	281	1.2	95.5				
ому	187	.8	96.3				

Frequency of Reference to Adjectival Dictionary Entries by Affix (Both Compatible and Incompatible) in Frequency Run V

TABLE E-3

Verbs							
Affix	Frequency	Percent	Cumulative Percentage	Affix	Frequency	Percent	Cumulative Percentage
ет	4353	19.5	19.5	ие	54	.2	97.8
ть	3008	13.5	33.0	ат	48	.2	98.0
я	2799	12.6	45.6	ами	47	.2	98.2
ют	1305	5.9	51.5	ю	45	.2	98.4
и	1269	5.7	57.2	ых	41	.2	98.6
ем	1170	5.3	62.5	ым	36		
ит	1068	4.8	67.3	ь	36		
им	1018	4.6	71.9	ах	29		
а	944	4.2	76.1	ев	28		
е	851	3.8	79.9	ые	24	.7	99.3
#	764	3.4	83.3	ое	21		
о	584	2.6	85.9	ыми	21		
ом	493	2.2	88.1	ям	21		
ут	414	1.9	90.0	ий	20		
ая	286	1.3	91.3	ями	20	.4	99.7
ы	274	1.2	92.5	ей	17		
ят	192	.9	93.4	ого	13		
ов	191	.9	94.3	ую	11		
у	151	.7	95.0	ам	10		
ой	129	.6	95.6	вши	8		
ая	119	.5	96.1	ый	6		
й	104	.5	96.6	ете	3		
ее	89	.4	97.0	ими	1		
ях	77	.3	97.3	ите	1		
в	58	.3	97.6	их	1	.3	100.0
					22265		

Frequency of Reference to Verbal Dictionary Entries by Affix (Both Compatible and Incompatible) in Frequency Run V

TABLE E-4

Appendix F

THE SUBROUTINES IN THE EXPERIMENTAL PREDICTIVE SYNTACTIC ANALYSIS PROGRAM

The explicit instructions for the operation of the experimental predictive syntactic analysis technique are presented in this appendix. Also included is a list of the function type and essence subroutines, as well as the PSI associated with each of the latter (Table F-1). The description of the different PSI is given in Table F-2. The abbreviations that are used in the main tables are listed in Table F-3, and in Table F-4, an outline of the format of the main tables is given.

The detailed operation of each subroutine is presented in Table F-5. An illustration of the use of this table will help familiarize the reader with the process. Consider the process when a subject prediction is being tested against the alternative argument, /noun, nominative, singular, masculine/ of a noun such as студэнт, the first word in a hypothetical sentence. The first entry under "Subject-E" in the table of essences signifies that the prediction was made either by initial, that is, at the beginning of a sentence, or by comma. If made by comma, the prediction is inactive initially, that is, its PSI = 51. It should also be noted that the subject prediction can be modified by a verb predicate head or an adjective predicate head if either of them precedes the subject.

The subject prediction can be fulfilled, on the one hand, by a noun, adjective, pronoun, or numeral alternative argument that is either nominative singular or nominative plural and, on the other hand, by an infinitive verb. Information in a reserved register indicates whether or not the predicate head has already been identified. If so, a word that can

be a subject is tested for agreement with it in person, number, and gender. In the example chosen, студент can be a nominative singular noun. Since it is the first word in the sentence, there is no information on person, number, or gender stored in the prediction pool, and студент fulfills the prediction.

If the preferred argument of the noun is subject, it is necessary to refer to the function type subroutines to determine what new predictions must be entered in the pool and what predictions already in the pool are to be modified (or wiped)

The noun function type subroutine first indicates the formal properties that identify a noun in the experimental program. The twelve essences that can be fulfilled by a noun are listed next, and, indeed, the Subject-E essence is among them. This function type subroutine is also called in by the pronoun and numeral function types.

The first prediction made by this subroutine for every noun alternative argument is for a noun complement. Since студент was selected as the subject, control is then transfered to the adjective-noun-subject function type subroutine.

The adjective-noun subject subroutine is accepted by nothing, that is, control is transfered to this subroutine only from another subroutine, in this case, the noun subroutine. The adjective-noun subject subroutine modifies the predicate head prediction if the latter has not been fulfilled previously. Next, three predictions are put into the prediction pool: compound subject, infinity, and end wipe. Since no other conditions are listed, control is transfered back to the skeleton which initializes the analysis of the next word.

For the reader who wishes to try the technique of predictive analysis, several Russian sentences analyzed on a word-by-word basis have been provided (Figs. F-1 to F-6). The grammar codes necessary to carry out the syntactic analysis are listed in Tables 3-4, 3-5, 3-7, 3-8, 3-9, and 3-12.

Essences (Tester Routines)	PSI	Function Types (Predictor Routines)
1. Subject - E	01	1. Initial
2. Compound Subject - E	99	2. Noun
3. Predicate Head	01	3. Pronoun
4. Compound Predicate Head	99	4. Adjective
5. Left Object - E	03	5. Numeral
6. Compound Left Object - E	99	6. Verb
7. Object - E	01	7. Adverb
8. Compound Object - E	99	8. Preposition
9. Master/(essence)	01	9. Participle
10. Verb Master - E	00	10. Gerund
11. Compound Verb Master - E	99	11. Infinite Conjunction
12. Noun Complement - E	00	12. Relative Conjunction - T
13. Compound Noun Complement - E	99	13. Comma
14. Preposition Complement - E	00	14. Adjective-Noun Subject
15. Compound Preposition		15. Pronoun Subject
Complement - E	99	16. Verb Subject
16. Phraser	03	17. Verb Predicate Head
17. Relative Conjunction - E	03	18. Adjective Predicate Head
18. Relative Pronoun - E	03	19. Left Object - T
19. Infinity	02	20. Object - T
20. End Wipe	01	21. Noun Complement - T
21. End of Sentence - E	01	22. Preposition Complement - T
22. Arbitrary Choice	02	23. Verb Master - T
		24. $ --- $
		25. End of Sentence - T

Index of Subroutines in the Experimental Predictive
Syntactic Analysis Program

TABLE F-1

PSI Value†		Description
00	-	Wiped by end wipe if not fulfilled by next word.
01	-	Wiped by end wipe. Must be fulfilled for analysis to be accepted, therefore, write on hindsight when wiped.
02	-	Wiped only by end wipe but not when fulfilled.
03	-	Wiped by end wipe.
49	-	Changed to 99 by prediction pool updating process; wiped by end wipe and end of sentence.
99	-	Inactive. Activated by infinite conjunction; wiped by end wipe and end of sentence.

†PSI + 50 - Inactivated (PSI = 99 is special case.)

Prediction Span Indicators (PSI)
in Experimental Predictive
Syntactic Analysis Program

TABLE F-2

Comp'd	- Compound
Compl.	- Complement
Subj.	- Subject
Obj.	- Object
Pred.	- Predicate
Prep'n	- Preposition
Conj.	- Conjunction
Adj.	- Adjective
PSI	- Prediction Span Indicator
OW	- Organized Word
CPx	- Character Position ($1 \leq x \leq 12$)
	$\boxed{1}\boxed{2}\boxed{3}\boxed{4}\boxed{5}\boxed{6}\boxed{7}\boxed{8}\boxed{9}\boxed{10}\boxed{11}\boxed{12}$
AWx	- Analyzed Word
x	= 1 : word 24 of 30 word item, or word 06 of texthadic item.
x	= 2 : word 27 of 30 word item, or word 07 of texthadic item.
TWx	- Texthadic Word ($0 \leq x \leq 9$)
GWx	- Grammatical Word (as kept in experimental program). ($1 \leq x \leq 5$)

List of Abbreviations

TABLE F-3

A. Essence
B. Predicted by: 1. 2. ⋮ List of all function types that predict the essence.
C. Modified by: 1. 2. ⋮ List of all function types that modify the essence.
D. Grammatical Information required: 1. 2. ⋮ Description of each word of grammatical information required by the tester routine in the order in which it must appear.
E. Fulfilled by: 1. 2. ⋮ List of function types recognized and the intersection test made with each.
F. Marks: XXXXXX Characteristic marking in TW9.

Format of Essence Table

TABLE F-4(a)

A. Function Type
B. Characterized by: 1. 2. ⋮ List of identifying symbols,
C. Accepted by: 1. 2. ⋮ List of names of various essences which accept this function type.
D. Called in by: 1. 2. ⋮ List of names of various function types which call in this function type.
E. Predicts: 1. 2. ⋮ List of all essences predicted by the routine in the order of their consideration, and/or list of all essences whose predictions are to be modified, with complete instructions for each modification.
F. Other conditions:

Format of Function Type Table

TABLE F-4(b)

ESSENCES	PREDICTED BY	MODIFIED BY	GRAMMATICAL INFORMATION REQUIRED
Subject-E	1. Initial (both active and inactive) 2. Comma (inactive)	1. Verb Pred. Head 2. Adj. Pred. Head	1. Person in CP1 of GW1 (V,Z,T,A) 2. Number in CP2 of GW1 (S,P,A) 3. Gender in CP1 of GW2 (M,F,N,A) 4. If CP3 of GW2 > 0, then Pred. Head has been found
Compound Subject-E	1. Adj.-Noun Subj. 2. Pronoun Subj. 3. Verb Subj.	1. Activated by Infinite Conj.	1. Person in CP1 of GW1 (V,Z,T,A) 2. Number in CP2 of GW1 (S,P,A) 3. Gender in CP1 of GW2 (M,F,N,A) 4. If CP2 of GW2 > 0, must be verb with F in CP9 of AW1 5. If CP3 of GW2 > 0, then Pred. Head has been found
Predicate Head	1. Initial (both active and inactive) 2. Comma (inactive)	1. Adj.-Noun Subj. 2. Pronoun Subj. 3. Verb Subj. 4. Left Obj.-T	1. Person in CP1 of GW1 (V,Z,T,A) 2. Number in CP2 of GW1 (S,P,A) 3. Gender in CP1 of GW2 (M,N,F, H,B,A) 4. If CP2 of GW2 > 0, an obj. has been found 5. If CP3 of GW2 > 0, a subj. has been found
Compound Pred. Head	1. Verb Pred. Head 2. Adj. Pred. Head	1. Activated by Infinite Conj.	1. Person in CP1 of GW1 (V,Z,T,A) 2. Number in CP2 of GW1 (S,P,A) 3. Gender in CP1 of GW2 (M,N,F, H,B,A)
Left Object-E	1. Initial (active and inactive) 2. Comma (inactive)	1. Wiped by Verb Pred. Head 2. Wiped by Adj. Pred. Head	none
Compound Left Object-E	1. Left Obj.-T	1. Activated by Infinite Conj.	1. Case word in appropriate position with zero fill
Object-E	1. Verb 2. Participle	nothing	1. Case and number word in appropriate positions with zero fill, see NAVI notation.
Compound Object-E	1. Obj.-T	1. Activated by Infinite Conj.	1. Case word in appropriate position with zero fill

Essence Subroutines Used in the Experimental Predictive Syntactic Analysis Program

TABLE F-5(a)

FULFILLED BY	MARKS
1. Noun 2. Adj. Satisfying grammatical information 3. Pronoun and N in CP1 or CP7 of AW1 4. Numeral 5. Verb with F in CP9 of AW1 satisfying grammatical information	1. ΔΔSUBJCTΔ
1. Noun 2. Adj. Satisfying grammatical information 3. Pronoun and N in CP1 or CP7 of AW1 4. Numeral 5. Verb with F in CP9 of AW1 with CP2 of GW2 > 0	1. ΔCSUBJCTΔ
1. Verb fulfilling grammatical information, with D in CP9 of AW1 2. Adj. Pred. Head, fulfilling grammatical information	1. ΔΔVΔPREDΔ 2. ΔΔAΔPREDΔ
1. Verb fulfilling grammatical information with D in CP9 of AW1 2. Adj. Pred. Head, fulfilling grammatical information	1. ΔCVΔPREDΔ 2. ΔCAΔPREDΔ
1. Noun 2. Pronoun With I in CP5 or CP11 of AW1 or 3. Adj. A in CP3 or CP9 of AW1, in that order 4. Numeral	1. ΔΔLΔOBJΔΔ
1. Noun 2. Pronoun with appropriate case in AW1 3. Adj. 4. Numeral	1. ΔCLΔOBJΔΔ
1. Noun 2. Adj. with appropriate case in AW1 3. Pronoun using NAVI notation 4. Numeral	1. ΔΔOBJECTΔ
1. Noun 2. Adj. with appropriate case in AW1 3. Pronoun 4. Numeral	1. ΔCOBJECTΔ

Table F-5(a) (continued)

ESSENCES	PREDICTED BY	MODIFIED BY	GRAMMATICAL INFORMATION REQUIRED
Master/ (essence)	1. Adj.	nothing	1. Unambiguous case and number with zero fill, using NAVI notation 2. Unambiguous gender with zero fill, using NAVI notation (M,F,N,A,B,U,H) 3. Mark of Essence which predicted the Master
Verb Master-E	1. Verb 2. Participle	nothing	none
Compound Verb Master-E	1. Verb Master-T	1. Activated by Infinite Conj.	none
Noun Complement-E	1. Noun	nothing	none
Compound Noun Complement-E	1. Noun Compl.-T	1. Activated by Infinite Conj.	none
Preposition Complement-E	1. Prep'n	nothing	1. Unambiguous case and number with zero fill in NAVI notation, i.e., if there is more than one case and number possibility, each one is considered as a separate prediction. Order of predictions: same as order of listing in SOW3.
Compound Preposition Complement-E	1. Prep'n Compl.-T	1. Activated by Infinite Conj.	1. Case of prep'n compl. in both singular and plural positions
Phraser	1. Comma 2. Initial	nothing	none

Table F-5(a) (continued)

FULFILLED BY		MARKS
1. Adj. 2. Noun 3. Pronoun 4. Numeral	With intersection in case, number, and gender in AW1 and AW2 using NAVI notation	1. XXXXXXXXM, where x-x is the marking of the essence of the word predicting the Master
1. Verb with F in CP9 of AW1		1. ΔΔVΔMASTΔ
1. Verb with F in CP9 of AW1		1. ΔCVΔMASTΔ
1. Adj. 2. Noun 3. Pronoun 4. Numeral	With G in CP2 or CP8 of AW1	1. ΔΔNΔCOMPΔ
1. Adj. 2. Noun 3. Pronoun 4. Numeral	With G in CP2 or CP8 of AW1	1. ΔCNΔCOMPΔ
1. Adj. 2. Noun 3. Pronoun 4. Numeral	With intersection in case in AW1 using NAVI notation	1. ΔΔRΔCOMPΔ
1. Noun 2. Adj. 3. Pronoun 4. Numeral	With intersection in case in AW1 using NAVI notation	1. ΔCRΔCOMPΔ
1. Participle 2. Verb with G in CP9 of AW1		1. ΔΔFRASERΔ

Table F-5(a) (continued)

ESSENCES	PREDICTED BY	MODIFIED BY	GRAMMATICAL INFORMATION REQUIRED
Relative Conjunction-E	1. Comma 2. Initial	nothing	none
Relative Pronoun-E	1. Comma 2. Initial	nothing	none
Infinity	1. Initial (inactive) 2. Comma (inactive) 3. Participle 4. Gerund 5. Pronoun Subj. 6. Adj.-Noun Subj. 7. Verb Subj. 8. Obj.-T 9. Left Obj.-T 10. Noun Compl. 11. Prep'n Compl.-T 12. Comma (twice) 13. Initial (four times)	nothing	none
End Wipe	1. Gerund 2. Participle 3. Comma (inactive) 4. Initial 5. Pronoun Subj. 6. Adj.-Noun Subj. 7. Verb Subj. 8. Obj.-T 9. Left Obj.-T 10. Noun Compl.-T 11. Prep'n Compl.-T 12. Comma (twice) 13. Initial (PSI=03) (three times)	1. Activated by Relative Pronoun-E 2. Activated by Relative Conj.-T	none
End of Sentence-E	1. Initial	nothing	none
Arbitrary Choice	1. Initial	nothing	none

Table F-5(a) (continued)

FULFILLED BY	MARKS
1. Relative Conj.-T	1. ΔΔRΔCONJΔ
1. Every Relative Pronoun fulfills this essence, whether or not there has been previous success. Upon fulfillment, the routine activates all predictions in the prediction pool with 50 ≤ PSI ≤ 98. It does not call to the success control routine, and continues as if there had been no success	1. Does not mark
1. Prep'n 2. Adverb 3. Dollar Sign 4. Comma 5. Infinite Conj.	1. INFΔPREPΔΔΔΔ 2. INFΔADVBΔΔΔΔ 3. INFΔ$$$$ΔΔΔΔ 4. INFΔCOMMAΔΔΔ 5. INFΔCONJNCTΔ
1. "No success" Wipes everything preceding in prediction pool including itself and continues down prediction pool. Writes all wiped PSI = 01 predictions on Hindsight tape	Does not mark
1. End of Sentence-T	1. ENDΔOFΔSENT. if . in CPl of OW 2. SEMICOLONΔΔ if ; in CPl of OW
1. Adj. 2. Noun 3. Pronoun 4. Verb 5. Numeral 6. Others not accepted by Infinity or Regular Essences, which do not make predictions	1. ΔΔΔRBTRΔΔ Note: Increase CHAIN by one.

Table F-5(a) (continued)

FUNCTION TYPE	CHARACTERIZED BY	ACCEPTED BY	CALLED IN BY
Initial	nothing	nothing	1. Program Initializer 2. End of Sentence-T
Noun	1. N in CP1 of OW 2. N in CP2 of OW, if P in GP1 of OW	1. Master/(essence) 2. Prep'n Compl. 3. Noun Compl. 4. Subj.-E 5. Left Obj.-E 6. Obj.-E 7. Comp'd Subj.-E 8. Comp'd Obj.-E 9. Comp'd Noun Compl.-E 10. Comp'd Left Obj.-E 11. Comp'd Prep'n Compl.-E 12. Arbitrary Choice	1. Pronoun 2. Numeral
Pronoun	1. P in CP1 of OW	1. Prep'n Compl.-E 2. Noun Compl.-E 3. Subj.-E 4. Obj.-E 5. Left Obj.-E 6. Comp'd Subj.-E 7. Comp'd Obj.-E 8. Comp'd Noun Compl.-E 9. Comp'd Left Obj.-E 10. Comp'd Prep'n Compl.-E 11. Arbitrary Choice	nothing

Function Type Subroutines Used in the Experimental Predictive Syntactic Analysis Program

TABLE F-5(b)

PREDICTS	OTHER CONDITIONS
1. Phraser 2. Infinity 3. End Wipe PSI = 03 4. Relative Conj. 5. Infinity 6. End Wipe PSI = 03 7. Relative Pronoun-E 8. Subj.-E (inactive) 9. Left Obj.-E (inactive) 10. Pred. Head (inactive) 11. Infinity (inactive) 12. End Wipe PSI = 03 13. Subj.-E 14. Left Obj.-E 15. Pred. Head 16. Infinity 17. End Wipe 18. End of Sentence-E 19. Infinity 20. Arbitrary Choice	1. Sets chain number to 00
1. Noun Compl.	If not master, and: 1. if subj. or comp'd subj., go to (a) Adj.-Noun Subj. or (b) Pronoun Subj. 2. if obj. or comp'd obj., go to Obj.-T 3. if left obj. or comp'd left obj., go to Left Obj.-T 4. if noun compl. or comp'd noun compl., go to Noun Compl.-T 5. if prep'n compl. or comp'd prep'n compl., go to Prep'n Compl.-T
nothing	1. If N in CP2 of OW, go to Noun 2. If A in CP2 of OW, go to Adj.

TABLE F-5(b) (continued)

FUNCTION TYPE	CHARACTERIZED BY	ACCEPTED BY	CALLED IN BY
Adjective	1. A in CP1 of OW, also CP9 of OW < 1 2. A in CP2 of OW, if P in CP1 of OW	1. Master/(essence) 2. Prep'n Compl. 3. Noun Compl. 4. Subj.-E 5. Obj.-E 6. Left Obj.-E 7. Comp'd Subj.-E 8. Comp'd Obj.-E 9. Comp'd Noun Compl.-E 10. Comp'd Left Obj.-E 11. Comp'd Prep'n Compl.-E 12. Arbitrary Choice	1. Pronoun
Numeral	1. D in CP1 of OW	1. Master/(essence) 2. Prep'n Compl.-E 3. Noun Compl.-E 4. Subj.-E 5. Obj.-E 6. Left Obj.-E 7. Comp'd Subj.-E 8. Comp'd Obj.-E 9. Comp'd Noun Compl.-E 10. Comp'd Left Obj.-E 11. Comp'd Prep'n Compl.-E 12. Arbitrary Choice	nothing
Verb	1. V in CP1 of OW	1. Pred. Head 2. Subj.-E 3. Verb Master-E 4. Phraser 5. Comp'd Subj.-E 6. Comp'd Pred. Head 7. Comp'd Verb Master-E 8. Arbitrary Choice	nothing
Adverb	1. H in CP1 of OW 2. A in CP1 of OW and 2 or 3 in CP9 of OW 3. A in CP1 of OW and 1 in CP8 of OW	1. Infinity	nothing
Preposition	1. R in CP1 of OW	1. Infinity	1. Prep'n (See Other Conditions)

TABLE F-5(b) (continued)

PREDICTS	OTHER CONDITIONS
1. Master/(essence)	If not master, and: 1. if subj. or comp'd subj., go to Adj.-Noun Subj. 2. if obj. or comp'd obj., go to Obj.-T 3. if left obj. or comp'd left obj., go to Left Obj.-T 4. if noun compl. or comp'd noun compl., go to Noun Compl.-T 5. if prep'n compl. or comp'd prep'n compl., go to Prep'n Compl.-T
1. If A in CP2 of OW: (a) Master/(essence) with PSI=00 with case and number determined by position (using NAVI Notation) in TW8, of any gender 2. If N in CP2 of OW: (a) nothing	1. If A in CP2 of OW, and not master: (a) if subj. or comp'd subj., go to Adj.-Noun Subj. (b) if obj. or comp'd obj. go to Obj.-T (c) if left obj. or comp'd left obj., go to Left Obj.-T (d) if noun compl. or comp'd noun compl., go to Noun Compl.-T (e) if prep'n compl. or comp'd prep'n compl., go to Prep'n Compl.-T 2. If N in CP2 of OW: (a) go to Noun
1. Verb Master-E 2. Obj.-E (a) if R in CP10 of AW1, then only Instrumental (b) if no government prediction in OW then only Accusative (c) if Pred. Head, and left obj. has been found, do not predict Obj.-E	1. If subj. or comp'd subj., go to Verb Subj. 2. If pred. head or comp'd pred. head, go to Verb Pred. Head 3. If G in CP9 of AW1, go to Gerund 4. If verb master or comp'd verb master, go to Verb Master-T
nothing	1. Inhibits wiping of Prediction Pool
1. Prep'n Compl.	1. Calls back to itself for making more than one unique prediction of prep'n compl.

TABLE F-5(b) (continued)

FUNCTION TYPE	CHARACTERIZED BY	ACCEPTED BY	CALLED IN BY
Participle	1. A in CP1 of OW and > 0 in CP10 of OW and not > 0 in CP8 and CP9 of OW	1. Phraser	nothing
Gerund	1. G in CP9 of AW1 and V in CP1 of OW	nothing	1. Verb
Infinite Conjunction	1. "и" and "или"	1. Infinity	nothing
Relative Con-junction-T	1. C in CP1 of OW; if "и" or "или" check prediction pool for unfulfilled Subj., Pred. Head, or Obj. If none, accept, otherwise reject.	1. Relative Conj.-E	nothing
Comma	1. , in CP1 of OW	1. Infinity	nothing
Adjective-Noun Sub-ject	1. i-iSUBJCTA in TW9 and neither V in CP1 of OW nor PN in CP1 and 2 of OW	nothing	1. Noun 2. Adj.

TABLE F-5(b) (continued)

PREDICTS	OTHER CONDITIONS
1. Verb Master 2. Obj.-E (government in CP1 and 2 of SOW3; if no P-code, accusative Obj. predicted) 3. Infinity 4. End Wipe	none
1. Infinity 2. End Wipe	none
nothing	1. After normal wiping of prediction pool, activate all predictions for which PSI = 99.
1. Activates all inactive predictions $(50 \leq (PSI) \leq 98)$	none
1. Phraser 2. Infinity 3. End Wipe PSI = 03 4. Relative Conj. 5. Infinity 6. End Wipe PSI = 03 7. Relative Pronoun-E 8. Subj.-E (inactive) 9. Left Obj.-E (inactive) 10. Pred. Head (inactive) 11. Infinity active (03) 12. End Wipe active (03)	1. Before making predictions, wipe all predictions in pool with $50 \leq PSI \leq 98$.
1. Modifies Pred. Head (if it has not been fulfilled) to 3rd person, and to number and gender of selected function and puts > 0 in CP3 of GW2. If comp'd subj., modify to 3rd person plural any gender. See "Pred. Head" for format information. 2. Comp'd Subj.-E with any person, number, and gender. 3. Infinity 4. End Wipe	none

TABLE F-5(b) (continued)

FUNCTION TYPE	CHARACTERIZED BY	ACCEPTED BY	CALLED IN BY
Pronoun Subject	1. i-iSUBJCTΔ in TW9 and PN in CP1 and 2 of OW	nothing	1. Noun
Verb Subject	1. i-iSUBJCTΔ in TW9 and V in CP1 of OW	nothing	1. Verb
Verb Predicate Head	1. i-iVΔPREDΔ in TW9	nothing	1. Verb
Adjective Predicate Head	1. A in CP1 of OW and 1 or 2 in CP9 of OW	1. Pred. Head	nothing
Left Object-T	1. i-iLΔOBJΔΔ in TW9	nothing	1. Noun 2. Adj.
Object-T	1. i-iOBJECTΔ in TW9	nothing	1. Noun 2. Adj.

TABLE F-5(b) (continued)

PREDICTS	OTHER CONDITIONS
1. Modifies Pred. Head (if it has not been fulfilled) as to person, number, and gender of pronoun and puts > 0 in CP3 of GW2. If comp'd subj., modify to 3rd person plural. See "Pred. Head" for format information. 2. Comp'd Subj.-E with any person, number, and gender. 3. Infinity 4. End Wipe	none
1. Modifies Pred. Head (if it has not been fulfilled) to 3rd person, neuter, singular, and puts > 0 in CP3 of GW2. See "Pred. Head" for format information 2. Comp'd Subj.-E with infinitive 3. Infinity 4. End Wipe	none
1. Modifies Subj.-E (if it has not been fulfilled) as to person, number, gender and puts > 0 in CP3 of GW2. See "Subj.-E" for format information 2. Erases Left Obj.-E, if it has not been fulfilled 3. Comp'd Pred. Head with person, number, and gender same as Verb Pred. Head	none
1. Comp'd Pred. Head - with person, number, and gender same as Adj. Pred. Head 2. Erases Left Obj. if it has not been fulfilled 3. Put > 0 in CP3 of GW2 of Subj.-E if not fulfilled	none
1. Puts > 0 in CP2 of GW2 of Pred. Head 2. Comp'd Left Obj.-E with same case as Left Obj.-T 3. Infinity 4. End Wipe	none
1. Comp'd Obj.-E with same case as Obj.-T 2. Infinity 3. End Wipe	none

TABLE F-5(b) (continued)

FUNCTION TYPE	CHARACTERIZED BY	ACCEPTED BY	CALLED IN BY
Noun Complement-T	1. i-iNΔCOMPΔ in TW9	nothing	1. Adj. 2. Noun
Preposition Complement-T	1. i-iRΔCOMPΔ in TW9	nothing	1. Adj. 2. Noun
Verb Master-T	1. i-iVΔMASTΔ in TW9	nothing	1. Verb
$ ---- $	1. $ in CPl of Russian word	1. Infinity	nothing
End of Sentence-T	1. . in OWl, CPl 2. ; in OWl, CPl	1. End of Sentence-E	nothing

TABLE F-5(b) (continued)

PREDICTS	OTHER CONDITIONS
1. Comp'd Noun Compl.-E 2. Infinity 3. End Wipe	none
1. Comp'd Prep. Compl.-E with same case 2. Infinity 3. End Wipe	none
1. Comp'd Verb Master-E	none
nothing	1. Inhibits wiping of prediction pool.
nothing	1. Wipes prediction pool completely. Put space blockette on Output 2. Goes to Initial

TABLE F-5(b) (continued)

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	3rd SEMI-OPGANIZED WORD	DICTIONARY SERIAL NO
CHARACTERIZE	V03.00	XARAKTERIZU- JA	00A-1128	VK OP30000	B G-	B1B2B4B5	211140000000
DEVICE	N01.00	PRIBOR-Y	00A-1129	NDI1M000	------N-A--- ------M-M---		157740000000
WIDTH	N04.00	SHIRIN-OJ	00A-1130	NDI2F000	----I------- ----F-------		214500000000
THEIR	P01.00	-IX	00A-11311	PA K ATS 0	NGACIPNGACIP AAAAAAAAAAAA		000115000000
THEM	P01.00	-IX	00A-1131%	PN K PTP 0	--------GA--- -------AA---		000110000000
BAND	N04.00	POLOS-Y	00A-1132	NDI2F000	-G----N-A--- -F----F-F---		150510000000
PASSAGE	N10.00	PROPUSKANI-J A	00A-1133	NDI1N100	-G----N-A--- -N----N-N---		163340000000
*.		*.	00A-1134	*			
WE	P01.00	M-Y	00A-1135	PN A PVP 0	------N----- ------H-----		102313333338
TO SUPPOSE	V01.00	PREDPOLAGA-E M	00A-1136	VN OP70000	---V--BAD-	B0B1B4B6	155590000000
EVEN	A02.00	RAVNOMERN-OF	00A-1137	AD00000	N-A--------- N-N---------		166890000000
DISTRIBUTION	N10.00	RAZPREDELENI -E	00A-1138	NDI1N000	N-A--------- N-N---------		171770555555
NOISE	N01.00	SHUM-A	00A-1139	NDI1M000	-G---------- -M----------		215870000000
ON	I01.00	P-O	00A-1140	R	--AC-P--AC-P	CAPROOF22611	135910000000
FREQUENCY	N04.00	CHASTOT-AM	00A-1141	NDI2F000	---------C-- ----------F--		212960000000
*.		*.	00A-1142	*			

A Sample Sentence Analyzed Word-by-word

Fig. F-1

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS	3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO
STARTING	N01.10	ZAPUSK-	00A-3010	NDI1M000	N-A--------- M-M---------		067290000000
FIRST	A03.00	PERV-OGO	00A-3011	AD00000	-GA--------- -BM---------		137380000000
MULTIVIBRATO R	N01.00	MUL.TIVIBRAT OR-A	00A-3012	NDI1M000	-G---------- -M----------		110670000000
CHAIN	N08.30	TSEPOCHK-I	00A-3013	NDI1F000	-G----N-A--- -F----F-F---		212300000000
AND	I01.00	-I	00A-3014	CH			000080000000
RETURN	N10.00	VOZVRASHCHEN I-E	00A-3015	NDI1N000	N-A--------- N-N---------		020830000000
ALL	P01.00	VSEX-	00A-3016	PK K PTF 0	--------GA--P -------AA--A		027685000000
TEN	D01.00	DESJAT-I	00A-3017	D RACJPK	-G-C-P-G-C-P -A-A-A-A-A-A	0000000TOTOT	049266666666
MULTIVIBRATO R	N01.00	MUL.TIVIBRAT OR-OV	00A-3018	NDI1M000	-------G---- -------M----		110670000000
IN(TO)	I01.00	-V	00A-3019	R	--A--P--A--P	APOROOBA0650	000020000000
INITIAL	A02.00	ISXODN-OE	00A-3020	AD00000	N-A--------- N-N---------		084360000000
POSITION	N10.00	POLOZHENI-E	00A-3021	NDI1N000	N-A--------- N-N---------		150410000000
TO CARRY OUT	V01.00	OSUSHCHESTVL JA-JUTSJA	00A-3022	VN OP70000	-----TBADR	B0B1B4B6	130380000000
WITH	I01.00	S-	00A-3023	R	-GA-I--GA-I-	GIAROOBA1111	178910000000
HELP	N06.10	POMOSHCH-.JU	00A-3024	NDI1F100	----I------- ----F-------		151710000000
GOVERNING	A04.00	UPRAVLJAJUSH CH-EJ	00A-3025	AD0100 4	-G-CIP------ -F-FFF------		204530000000
CIRCUIT	N04.00	SXEM-Y	00A-3026	NDI2FOYO	-G----N-A--- -F----F-F---		194360000000
*.		*.	00A-3027	*			

A Sample Sentence Analyzed Word-by-word

Fig. F-2

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
AUTONOMOUS	A02.00	SAMOSTOJATEL .N-OE	00A-2901	AD00000	N-A----------	N-N----------		179790000000
RETURN	N10.00	VOZVRASHCHEN I-E	00A-2902	NDI1N000	N-A----------	N-N----------		020830000000
MULTIVIBRATO R	N01.00	MUL.TIVIBRAT CR-OV	00A-2903	NDI1M000	--------G----	--------K----		110670000000
IN/TO)	I01.00	-V	00A-2904	R	--A--P--A--P		APOR00BA0650	000020000000
INITIAL	A02.00	ISXODN-OE	00A-2905	AD00000	N-A----------	N-N----------		084360000000
STATE	N10.00	SOSTOJANI-E	00A-2906	NDI1N100	N-A----------	N-N----------		189220000000
TO AVERT	V01.00	PREDOTVRASHC HA-ETSJA	00A-2907	VN OP70000	--T---BADR			155460000000
SELECTION	N01.00	VYBOR-OM	00A-2908	NDI1M000	----I--------	----M--------	B0B1B4B6	030380000000
SUFFICIENT	A02.00	DOSTATOCHN-O	00A-29091	AD00000 2	N------------	N------------		055150000000
ENOUGH	D01.00	DOSTATOCHN-O	00A-29092	D XEACUNY S	N-A---N-A---	A-A---A-A---	0000000T0000	055145000000
IT IS ENOUGH	I01.00	DOSTATOCHN-O	00A-2909%	X				055140000000
BIG	A08.00	BOL.SH-OJ	00A-2910	AD01000	NGACIP------	MFMFFF------		009100000000
CONSTANT	A01.00	POSTOJANN-OJ	00A-2911	KDIOFO	-G-CIP------	-F-FFF------	T6	152970000000
TIME	N12.00	VREMEN-I	00A-2912	NDI1N000	-G-C-P------	-N-N-N------		027060000000
CIRCUIT	N06.00	TSEP-I	00A-2913	NDI1FOYO	-G-C-PN-A---	-F-F-FF-F---		212260000000
GRID	N04.31	SETK-I	00A-2914	NDI1F000	-G----N-A---	-F----F-F---		183140000000
RIGHT	A03.00	PRAV-OGO	00A-2915	AD00000	-GA----------	-BM----------		154280000000
TRIODE	N01.00	TRIOD-A	00A-2916	NDI1M000	-G-----------	-M-----------		200830000000
MULTIVIBRATO R	N01.00	MUL.TIVIBRAT OR-A	00A-2917	NDI1M000	-G-----------	-M-----------		110670000000
*.		*.	00A-2918	*				

A Sample Sentence Analyzed Word-by-word

Fig. F-3

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
BEHIND	I01.00	Z-A	00A-0202	R	--A-I---A-I-		IA0R00BG0680	057200000000
LAST	A05.00	POSLEDN-EE	00A-0203	AD010001	N-A----------	N-N----------		152560000000
TIME	N12.00	VREM-JA	00A-0204	NDI1N000	N-A----------	N-N----------		027040000000
OFFERED	A01.00	PREDLOZHEN-O	00A-0205	AD0000 23	N------------	N------------		154960000000
A FEW	D01.00	NESKOL.K-O	00A-0206	D XEACUNYKK	N-A---N-A---	A-A---A-A---	0000000T0000	118950333328
METHOD	N01.00	SPOSOB-OV	00A-0207	NDI1M000	--------G----	--------M----		190300000000
AVERAGING	N10.00	USREDNENI-JA	00A-0208	NDI1N000	-G----N-A---	-N----N-N---		206195000000
SIGNAL	N01.00	SIGNAL-OV	00A-0209	NDI1M000	--------G----	--------K----		183370000000
ARBITRARY	A02.00	PROIZVOL.N-O J	00A-0210	AD00000	-G-CIP------	-F-FFF------		162400000000
FORM	N04.00	FORM-Y	00A-0211	NDI2F000	-G----N-A---	-F----F-F---		209340000000
*.		*.	00A-0212					

A Sample Sentence Analyzed Word-by-word

Fig. F-4

FIRST ENGLISH EQUIVALENT	CLASS MARKER	RUSSIAN WORD (TRANSLITERATED)	TEXT SERIAL NO.	ORGANIZED WORD	CODING DUE TO WORD-BY-WORD ANALYSIS		3rd SEMI-ORGANIZED WORD	DICTIONARY SERIAL NO.
ESSENTIAL	A01.00	SUSHCHESTVEN N-O	00A-1045	AD01000 2	N----------	N----------		194060000000
*.		*.	00A-1046	*				
THAT	I01.00	CHT-O	00A-10471	C				213848750000
WHICH	P01.00	CHT-O	00A-1047*	PNCI STRI O	N-A---------	N-N---------		213847500000
FOR	I01.00	DL-JA	00A-1048	R	-G-----G----		G00R00600400	051970000000
GIVEN	A01.00	ZADANN-OGO	00A-1049	AD0000 3	-GA---------	-BM---------		060932500000
TIME	N12.00	VREMEN-I	00A-1050	NDI1N000	-G-C-P------	-N-N-N-----		027060000000
ESTABLISHMEN T	N10.00	USTANOVLENI- JA	00A-1051	NDI1NOYO	-G----N-A---	-N-----N-N---		206560000000
READING	N10.00	POKAZANI-J	00A-1052	NDI1N000	--------G----	-------N---		149310000000
DEVICE	N01.00	PRIPOR-A	00A-1053	NDI1M000	-G----------	-M---------		157740000000
AVERAGE	A05.00	SREDN-IJ	00A-1054	AD01000	N-A---------	M-M---------		190910000000
SQUARE	N01.00	KVADR-AT	00A-1055	NDI1M000 1	N-A---------	M-M---------		087120000000
FLUCTUATION	N07.00	FLUKTUATSI-I	00A-1056	NDI1F000	-G-C-PN-A---	-F-F-FF-F---	T5	208760000000
RESULT	N01.00	REZUL.TAT-A	00A-1057	NDI1M000	-G----------	-M---------		176530000000
MEASUREMENT	N10.00	IZMERENI-JA	00A-1058	NDI1N000	-G----N-A---	-N-----N-N---		076520000000
DEPEND	V06.20	ZAVIS-IT	00A-1059	VN OF40000	--T---BAD-		B1	058630000000
FROM	I01.00	OT-	00A-1060	R	-G-----G----		G00R00D00400	130520000000
GREATER	A04.00	BOL.SH-EJ	00A-1061	AD00000	-G-CIP------	-F-FFF------		009095000000
OR	I01.00	IL-I	00A-1062	C				078400000000
LESSER	A04.00	MEN.SH-EJ	00A-1063	KDKOM00	-G-CIP------	-F-FFF------	T7	105670000000
DETAIL	N06.00	DETAL.NOST-I	00A-1064	NDI1F000	-G-C-PN-A---	-F-F-FF-F---		049505000000
RECORDING	N07.00	REGISTRATSI- I	00A-1065	NDI1F000	-G-C-PN-A---	-F-F-FF-F---		175835000000
SIGNAL	N01.00	SIGNAL-A	00A-1066	NDI1M000	-G----------	-M---------		183370000000
*.		*.	00A-1067	*				
WHICH	P01.00	KOTOR-AJA	00A-1068	PK K STRITO	N-----------	F-----------		095057959183
CHARACTERIZE	V03.00	XARAKTERIZU- ETSJA	00A-1069	VK OP30000	--T---XADR		B1B2B4B5	211140000000
TIME	N12.00	VREMEN-EM	00A-1070	NDI1N000	-----I------	----N-------		027060000000
$DELTA T		$DELTA T	00A-1071	###				
*.		*.	00A-1072	*				

A Sample Sentence Analyzed Word-by-word

Fig. F-5